

Jason Hodge

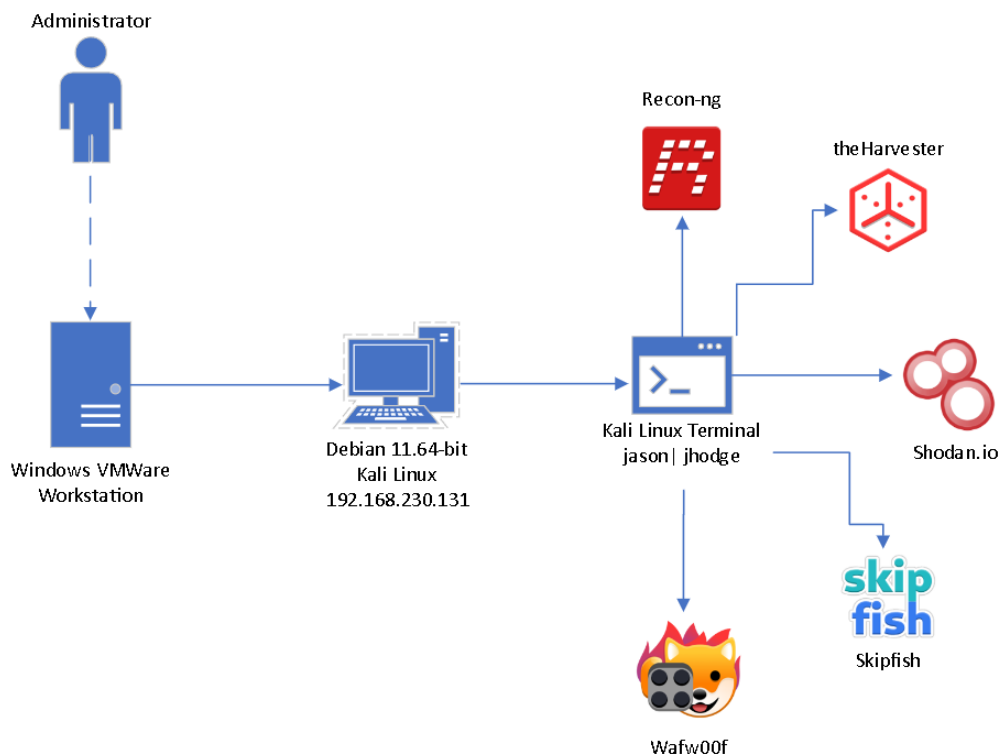
Lab 03 – Open Source Intelligence Gathering

October 25, 2023

Description:

In the first part of this lab, I dove into researching my target organization Marist College. Then, through utilizing free and Open-Source Intelligence (OSINT) framework tools, I was able to find quite the trove of valuable information to conduct penetration tests with Kali Linux tools about the Marist domain. Then through using passive reconnaissance tools I discovered more information about Marist College including Ip addresses, firewalls, subdomains, hosts, networks, etc. Passive tools utilized in this lab include Recon-ng, TheHarvester, Shodan.io, Skipfish, and Wafw00f. All five of these tools are information gathering tools used to find potential vulnerability areas to assist with building an overall profile of a target's security. With the found information I conducted an analysis of each tool and its importance.

Topology:



This is an overview of the entire lab and the passive open-source Kali tools used within my Debian Kali Linux machine on VMWare Workstation.

Key Syntax:

- The name of the Kali tool you are using always goes before the command you are running.
- Various commands to download and run all of these tools are included throughout the report.

Verification:

TASK ONE: Passive Discovery

Some public information I discovered through gathering Open Source Intelligence (OSINT)(OSINT framework) from using Google include:

- Various Marist email addresses with the format firstname.lastname1@marist.edu
- Marist standard phone number is 845-575-3000
 - A whole directory of phone numbers: <https://www.marist.edu/directory>
- Marist Instagram, Facebook, Twitter(X), LinkedIn, TikTok, YouTube, Pinterest.
- Many web pages: <https://www.marist.edu/about/marist-at-a-glance>,
<https://maristpoll.marist.edu/>, <https://www.marist.edu/marist100>,
<https://www.marist.edu/student-life/campus-services>, etc.
- The domain is <https://www.marist.edu/>

Information below was found utilizing Whois.com:

- Domain Name: MARIST.EDU
- Registrant:
 - Marist College
 - 3399 North Road
 - Poughkeepsie, NY 12601
 - USA
- Administrative Contact:
 - A Williams
 - Marist College
 - North Road
 - Poughkeepsie, NY 12601-1387
 - USA
 - +1.8455753252
 - email@vm.marist.edu

- Technical Contact:
A Williams
Marist College
North Road
Poughkeepsie, NY 12601-1387
USA
+1.8455753252
email@vm.marist.edu
- Name Servers:
NS1.MARIST.EDU
NS3.MARIST.EDU
NS4.MARIST.EDU
NS2.MARIST.EDU
- Domain record activated: 31-Aug-1989
- Domain record last updated: 07-Jul-2023
- Domain expires: 31-Jul-2024
- IP Address ranges 48.100.172.0 - 148.100.172.255 and 148.100.131.0 - 148.100.131.255
(db ip)

Recon-ng: A reconnaissance framework that is useful to gain opensource intelligence for vulnerability assessment regarding hosts and IP addresses, if applicable, on a domain.

```
jason@jhodge: ~  
File Actions Edit View Help  
[jason@jhodge]~$ recon-ng  
[*] Version check disabled.  
  
Sponsored by ...  
www.blackhillsinfosec.com  
  
PRACTISEC  
www.practisec.com  
  
[recon-ng v5.1.2, Tim Tomes (@lanmaster53)]  
  
[*] No modules enabled/installed.  
  
[recon-ng][default] > marketplace install all  
[*] Module installed: discovery/info_disclosure/cache_snoop  
[*] Module installed: discovery/info_disclosure/interesting_files  
[*] Module installed: exploitation/injection/command_injector  
[*] Module installed: exploitation/injection/xpath_bruter  
[*] Module installed: import/csv_file  
[*] Module installed: import/list  
[*] Module installed: import/masscan  
[*] Module installed: import/nmap  
[*] Module installed: recon/companies-contacts/bing_linkedln_cache  
[*] Module installed: recon/companies-contacts/censys_email_address  
[*] Module installed: recon/companies-contacts/pen  
[*] Module installed: recon/companies-domains/censys_subdomains  
[*] Module installed: recon/companies-domains/pen  
[*] Module installed: recon/companies-domains/viewdns_reverse_whois  
[*] Module installed: recon/companies-domains/whoxy_dns  
[*] Module installed: recon/companies-hosts/censys_org  
[*] Module installed: recon/companies-hosts/censys_tls_subjects  
[*] Module installed: recon/companies-multi/github_miner  
[*] Module installed: recon/companies-multi/shodan_org  
[*] Module installed: recon/companies-multi/whois_miner  
[*] Module installed: recon/contacts-contacts/abc  
[*] Module installed: recon/contacts-contacts/alltester  
[*] Module installed: recon/contacts-contacts/mangle  
[*] Module installed: recon/contacts-contacts/unmangle  
[*] Module installed: recon/contacts-credentials/hibp_breach  
[*] Module installed: recon/contacts-credentials/hibp_paste
```

First, in order to use Recon-ng and other Kali Linux environment tools I used the “marketplace install all” command to install all the package modules within Kali.

```
s/search/___init___py)'.
[recon-ng][default] > help

Commands (type [help?] <topic>):

back                Exits the current context
dashboard           Displays a summary of activity
db                 Interfaces with the workspace's database
exit               Exits the framework
help               Displays this menu
index              Creates a module index (dev only)
keys               Manages third party resource credentials
marketplace         Interfaces with the module marketplace
modules            Interfaces with installed modules
options            Manages the current context options
pdb                Starts a Python Debugger session (dev only)
script             Records and executes command scripts
shell              Executes shell commands
show               Shows various framework items
snapshots          Manages workspace snapshots
spool              Spools output to a file
workspaces          Manages workspaces

[recon-ng][default] > █
```

The help command, “-h” shows the available commands that could be utilized in a recon-ng scan.

```
[recon-ng][default] > modules search brute
[*] Searching installed modules for 'brute' ...

Exploitation
  exploitation/injection/xpath_bruter

Recon
  recon/domains-domains/brute_suffix
  recon/domains-hosts/brute_hosts

[recon-ng][default] > workspaces create L3
[!] 'google_api' key not set. pushpin module will likely fail at runtime. See 'keys add'.
[!] 'flickr_api' key not set. flickr module will likely fail at runtime. See 'keys add'.
[!] 'google_api' key not set. youtube module will likely fail at runtime. See 'keys add'.
[!] 'twitter_api' key not set. twitter module will likely fail at runtime. See 'keys add'.
[!] 'twitter_secret' key not set. twitter module will likely fail at runtime. See 'keys add'.
[!] 'shodan_api' key not set. shodan module will likely fail at runtime. See 'keys add'.
[!] Module 'recon/companies-hosts/censys_org' disabled. Dependency required: 'me 'CensysIPv4' f
it__.py)'.
[!] Module 'recon/companies-hosts/censys_tls_subjects' disabled. Dependency required: 'me 'Cens
arch/__init__.py)'.
[!] 'hibp_api' key not set. hibp_paste module will likely fail at runtime. See 'keys add'.
[!] 'hibp_api' key not set. hibp_breach module will likely fail at runtime. See 'keys add'.
[!] Module 'recon/netblocks-companies/censys_netblock_company' disabled. Dependency required: '
ensys/search/__init__.py)'.
[!] 'github_api' key not set. github_repos module will likely fail at runtime. See 'keys add'.

[recon-ng][L3] > modules load recon/domains-hosts/brute_hosts
[recon-ng][L3][brute_hosts] > options
Manages the current context options

Usage: options <list|set|unset> [...]

[recon-ng][L3][brute_hosts] > options list

  Name      Current Value      Required  Description
  -----
SOURCE      default              yes       source of input (see 'info' for details)
WORDLIST    /home/jason/.recon-ng/data/hostnames.txt  yes       path to hostname wordlist

[recon-ng][L3][brute_hosts] > options set SOURCE marist.edu
SOURCE ⇒ marist.edu
[recon-ng][L3][brute_hosts] > run
```

In this part, I used the “modules search brute” command to search for the brute module. I used the reconnaissance module “recon/domains-hosts/brute_hosts” to find list of subdomains on the Marist network. I then created a workspace called L3 to load and run the source Marist.edu in.

SUMMARY

```
[*] 221 total (126 new) hosts found.
[recon-ng][L3][brute_hosts] > show hosts
```

rowid	host	ip_address	region	country	latitude	longitude	notes	module
1	www.marist.edu							brute_hosts
2	admin.marist.edu							brute_hosts
3	www.ha.marist.edu							brute_hosts
4	admin.marist.edu	148.100.2.4						brute_hosts
5	autodiscover.outlook.com							brute_hosts
6	autodiscover.marist.edu							brute_hosts
7	atod-g2.tm-4.office.com							brute_hosts
8	autodiscover.marist.edu	52.96.109.184						brute_hosts
9	autodiscover.marist.edu	52.96.35.8						brute_hosts
10	autodiscover.marist.edu	52.96.69.8						brute_hosts
11	autodiscover.marist.edu	52.96.182.104						brute_hosts
12	autodiscover.marist.edu	52.96.111.40						brute_hosts
13	autodiscover.marist.edu	52.96.87.232						brute_hosts
14	autodiscover.marist.edu	52.96.9.184						brute_hosts
15	autodiscover.marist.edu	52.96.109.232						brute_hosts
16	marist.catalog.acalog.com							brute_hosts
17	catalog.marist.edu							brute_hosts
18	hera-c.aws.acalog.com							brute_hosts
19	acalog-production-dir-new-6ee7eb4fb0a2b456.elb.us-east-1.amazonaws.com							brute_hosts

This is part of the list of host subdomains found on the Marist network. We can see that 126 hosts were found. The “show hosts” command expanded the results for easier viewing.

```
[*] 126 rows returned
[recon-ng][L3][brute_hosts] > modules search reporting
[*] Searching installed modules for 'reporting'...
```

Reporting

- reporting/csv
- reporting/html
- reporting/json
- reporting/list
- reporting/proxifier
- reporting/pushpin
- reporting/xlsx
- reporting/xml

```
[recon-ng][L3][brute_hosts] > modules load reporting/html
[recon-ng][L3][html] > options
Manages the current context options

Usage: options <list|set|unset> [...]
```

```
[recon-ng][L3][html] > options list
```

Name	Current Value	Required	Description
CREATOR		yes	use creator name in the report footer
CUSTOMER		yes	use customer name in the report header
FILENAME	/home/jason/.recon-ng/workspaces/L3/results.html	yes	path and filename for report output
SANITIZE	True	yes	mask sensitive data in the report

```
[recon-ng][L3][html] > options set CREATOR J
CREATOR => J
[recon-ng][L3][html] > options set CUSTOMER Oops
CUSTOMER => Oops
```

```
[recon-ng][L3][html] > options set FILENAME /home/jason/Documents/Marist_Report.html
FILENAME => /home/jason/Documents/Marist_Report.html
[recon-ng][L3][html] > run
[*] Report generated at '/home/jason/Documents/Marist_Report.html'.
[recon-ng][L3][html] > █
```

In this last part I wanted to save the results of the test as an html file type. To do this I had to set a creator of the file, a customer, set the file save location, and name the file before generating the report.

Subdomains are susceptible to hijacking, which can be used for various purposes such as hurting companies' reputations, stealing users' data, various phishing scams, etc. Some best practices to prevent against threats like these include constantly updating subdomains, setting up proper firewalls, utilize honeypots to capture the attacker in an isolated environment and learn about them to reinforce the protective layers (HackerNoon)

TheHarvester: A passive reconnaissance tool used to gather names, email addresses, subdomains, hosts, and more from various search engines and other public resources. TheHarvester is normally used for the early stages of a penetration test to fully understand the customer or target being pursued.

```
(jason@jhodge)-[~]
$ sudo apt install theharvester
[sudo] password for jason:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  python3-texttable
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  theharvester
1 upgraded, 0 newly installed, 0 to remove and 1325 not upgraded.
Need to get 679 kB of archives.
After this operation, 67.6 kB of additional disk space will be used.
Get:1 http://kali.darklab.sh/kali kali-rolling/main amd64 theharvester all 4.4.3-0kali1 [679 kB]
Fetched 679 kB in 1s (741 kB/s)
(Reading database ... 392649 files and directories currently installed.)
Preparing to unpack .../theharvester_4.4.3-0kali1_all.deb ...
Unpacking theharvester (4.4.3-0kali1) over (4.2.0-0kali1) ...
Setting up theharvester (4.4.3-0kali1) ...
Installing new version of config file /etc/theHarvester/api-keys.yaml ...
Processing triggers for kali-menu (2023.1.7) ...
```

Here we can see the Kali Linux tool, theHarvester being installed on the machine with the command “sudo apt install theharvester”.

```

(jason@jhodge)-[~]
$ theHarvester -d marist.edu -b all -f Harvester_Report
*****
*
* theHarvester 4.4.3
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*
*****

[*] Target: marist.edu

[!] Missing API key for bevigil.
[!] Missing API key for binaryedge.
[!] Missing API key for bufferoverrun.
[!] Missing API key for Censys ID and/or Secret.

```

Before running any command the tool you are using needs to be initialized ahead of the scan or test you are about to run. In this test, the command “theHarvester -d marist.edu -b all -f Harvester_Report” identifies the Marist domain as what we are searching for and then searches information from all sources. The last part creates a report of the results as both JSON and XML files.


```

54.164.4.253
54.175.250.58
54.210.232.132
54.243.238.66
54.243.250.147
54.81.232.47
54.85.246.96
54.87.189.131
54.88.35.99
64.225.35.74
64.91.243.43
72.52.5.200
88.221.168.234

[*] Emails found: 16
admission@marist.edu
anthony.proia@marist.edu
deborah.holtman@marist.edu
graduate@marist.edu
helpdesk@marist.edu
international.studentservices@marist.edu
international@marist.edu
mc.cls@marist.edu
michelle.eggink@marist.edu
registrar@marist.edu
safety@marist.edu
studentfinancialservices@marist.edu
tracey.niemotko@marist.edu
transcript.request@marist.edu
veterans@marist.edu

[*] Hosts found: 13460
148-100-128-100.foxnet.marist.edu
148-100-128-101.foxnet.marist.edu
148-100-128-102.foxnet.marist.edu
148-100-128-104.foxnet.marist.edu
148-100-128-105.foxnet.marist.edu
148-100-128-108.foxnet.marist.edu
148-100-128-109.foxnet.marist.edu
148-100-128-11.foxnet.marist.edu

```

This is a preview of ip addresses, email addresses, and hosts discovered on Marist's network. 13460 is a lot of devices on the foxnet network.

```

xymon-remote.it.marist.edu:64.225.35.74
xymon.it.marist.edu
xymon.it.marist.edu:hobbit.it.marist.edu
xymon.it.marist.edu:hobbit.it.marist.edu.
xymontst.is.marist.edu
xynet.it.marist.edu
xynet.it.marist.edu:10.13.8.62
zcloud.marist.edu
zcloud.marist.edu:148.100.42.42
zos.kctr.marist.edu
zos.kctr.marist.edu:148.100.96.12
zos.kctr.marist.edu:vm.marist.edu
zsakai.marist.edu
zseries.marist.edu
zseries.marist.edu:cart.it.marist.edu
zseries.marist.edu:cart.it.marist.edu.
zvmclass.marist.edu

[*] Reporting started.
[*] XML File saved.
[*] JSON File saved.

```

```

(jason@jhodge)~$

```

Here we can see that the report was saved as both an XML file and JSON file to the scans folder on the machine.

The result of this test provides a list of Ip addresses, email addresses, and hosts that can all be targeted by various attacking methods such as phishing, ip spoofing, DDoS, malware, etc. These can all be prevented by using best practices such as not opening suspicious emails and changing passwords regularly, keeping software up to date, etc.

Shodan.io: The Shodan.io browser tool is used to find vulnerabilities in systems by regularly crawling the internet for information on network systems. This tool was used to search Marist for devices on its network and provides identifiable information about them.

SHODAN

Explore

Downloads

Pricing

hostname:"marist.edu"

Q

Account

TOTAL RESULTS

1,659

TOP PORTS

5000	442
7000	136
22	123
443	112
49152	98

More...

TOP ORGANIZATIONS

Marist College	1,648
Amazon Technologies Inc.	5

More...

TOP PRODUCTS

AirPlay	578
OpenSSH	146
Apache httpd	105
Apple remote desktop vnc	59
mDNS	43

More...

TOP OPERATING SYSTEMS

Mac OS X	59
----------	----

View Report

View on Map

Access Granted: Want to get more out of your existing Shodan account? Check out [everything you have access to.](#)

148.100.52.146

2023-10-18T20:15:18.814984

148-100-52-146.stu.ce.net.marist.edu

Marist College

United States, Fairview

SSH-2.0-OpenSSH_5.9

Key type: ssh-rsa

Key: AAAAB3NzaC1yc2EAAAADAQABAAQDLVTrvFPgAKM6H2pkMnhXNHaV9k0y7GcINW3Ck88G26Vqn201H35g8F/Q2/YfM1lokK7x2udk3ysvtbFx9IV8oFkDK49zJNAJ3pMbSn3IPaHrtYG9FZv8H2DEKIALatFTYeicVxktwtzm9ky9Latzo2HgSHNFEAN5zAhSAPg2X6gN3L1kb/7p9xaI70zS51JNg0X1XQYsm1epVGITi1hX4XtWjKaO...

Marist College

2023-10-18T20:05:22.548185

148.100.2.135

emr-dev.it.marist.edu

Marist College

United States, Poughkeepsie

SSL Certificate

HTTP/1.1 200 OK

Issued By: InCommon RSA Server CA

Date: Wed, 18 Oct 2023 20:05:22 GMT

Server: Apache

Content-Length: 2182

Content-Type: text/html

Marist College

2023-10-18T20:05:22.548185

148.100.2.135

emr-dev.it.marist.edu

Marist College

United States, Poughkeepsie

SSL Certificate

HTTP/1.1 200 OK

Issued By: InCommon RSA Server CA

Date: Wed, 18 Oct 2023 20:05:22 GMT

Server: Apache

Content-Length: 2182

Content-Type: text/html

Marist College

2023-10-18T19:59:32.836958

148.100.61.23

148-100-61-23.stu.dn.net.marist.edu

Marist College

United States, Poughkeepsie

HTTP/1.1 403 Forbidden

Content-Length: 0

Server: AirTunes/377.40.00

IoT

AirPlay:

Name: Basketball Room 217

Device Model: Roku 4660X

Hardware Revision: PVT1

Serial Number: d0be8e46-2384-5175-a226-dd5becd7d72a

TOP OPERATING SYSTEMS

Mac OS X	59
Windows	11
Ubuntu	8
Debian	2
Linux	2
More...	

Device Model: Roku 4660X
Hardware Revision: PVT1
Serial Number: d0be8e46-2384-5175-a226-dd5becd7d72a
SDK: AirPlay;3.4.2.8
Firmware Build: 40.00
Firmware Build Date: Mar 20 ...

148.100.144.249

148-100-144-249.FoxNet.marist.edu
Marist College
United States, Poughkeepsie



HTTP/1.1 403 Forbidden
Content-Length: 0
Server: AirTunes/690.7.1

2023-10-18T19:56:57.077903

AirPlay:
Name: Jill's MacBook Air
Device Model: Mac14,2
AirPlay Version: 690.7.1
Device ID: 1C:57:DC:27:B3:56
MAC Address: 1C:57:DC:27:B3:56
Protocol Version: 1.1

Access forbidden!

148.100.100.12
genweb2.it.marist.edu
presidential.marist.edu
Marist College
United States, Poughkeepsie

HTTP/1.1 403 Forbidden
Date: Wed, 18 Oct 2023 19:52:47 GMT
Server: Apache
Vary: accept-language,accept-charset
Accept-Ranges: bytes

2023-10-18T19:52:48.017976

Access forbidden!

148.100.100.12
genweb2.it.marist.edu
presidential.marist.edu
Marist College
United States, Poughkeepsie

HTTP/1.1 403 Forbidden
Date: Wed, 18 Oct 2023 19:52:47 GMT
Server: Apache
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
Content-Language: en

2023-10-18T19:52:48.017976

148.100.153.28

148-100-153-28.FoxNet.marist.edu
Marist College
United States, Poughkeepsie



HTTP/1.1 403 Forbidden
Content-Length: 0
Server: AirTunes/675.4.1

AirPlay:
Name: Basma's MacBook Pro
Device Model: MacBookPro17,1
AirPlay Version: 675.4.1
Device ID: A0:78:17:6B:47:AC
MAC Address: A0:78:17:6B:47:AC
Protocol Version: 1.1

2023-10-18T19:36:22.528237

148.100.49.200

portforward.marist.edu
Marist College
United States, Poughkeepsie

Marist College System

--MARIST --PRESS BREAK KEY TO BEGIN SESSION.

2023-10-18T19:29:57.646542

I ran the Shodan.io browser tool with the command `hostname:"marist.edu"`. With this command I was able to discover information about network devices on Marist's system. What was found pertains to host device information such as Ip address, location, type of device and model, etc. This is just a small portion of what was found.

Shodan Report

hostname:"marist.edu"

Total: 1,660

// GENERAL



Countries

United States	1,660
---------------	-------

Ports

5000	442
7000	137
22	123
443	112
49152	98

MORE...

Organization

Marist College	1,649
Amazon Technologies Inc.	5
DigitalOcean, LLC	3
Amazon.com, Inc.	2
Amazon Data Services NoVa	1

MORE...

Vulnerabilities

No information available.

Products

AirPlay	579
OpenSSH	140
Apache httpd	105
Apple remote desktop vnc	59
mDNS	43

MORE...

Tags

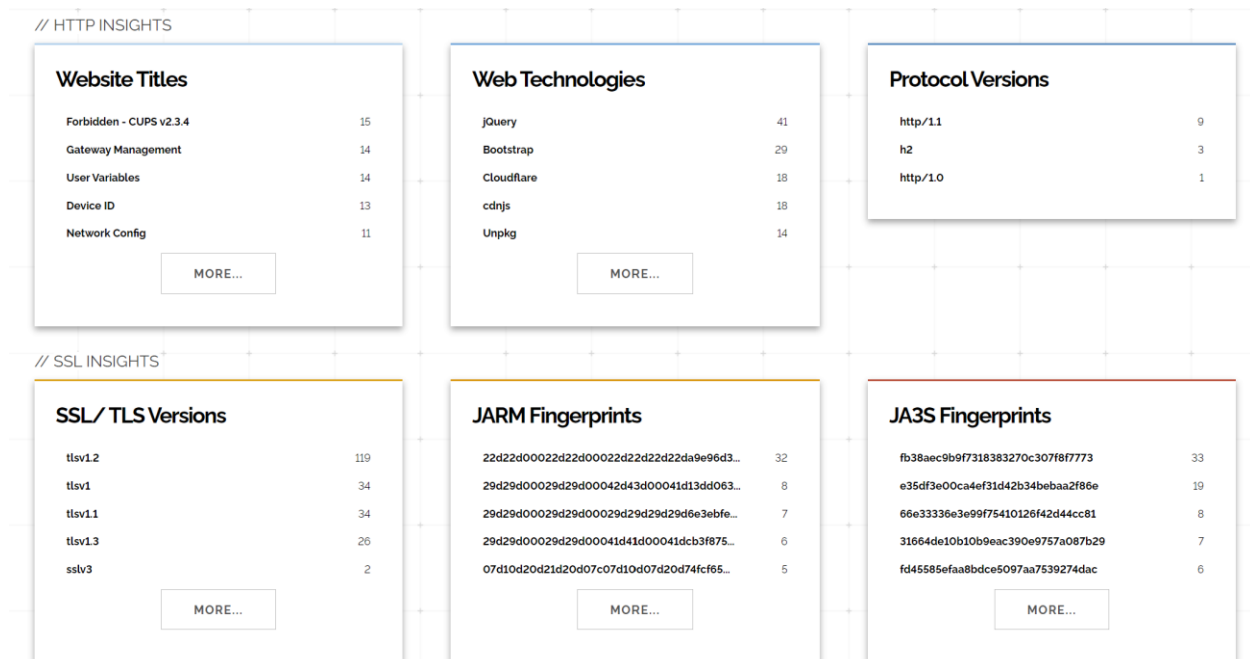
lot	582
database	31
self-signed	26
cloud	11
eol-product	5

MORE...

Operating Systems

Mac OS X	59
Windows	11
Ubuntu	8
Debian	2
Linux	2

MORE...



Here we see an overview of the Marist network search information discovered as a report form.

Skipfish: A passive reconnaissance tool used to gain information about system vulnerabilities where a hacker can gain entry to a system. The scan I used did not limit the crawling depth, by using the -o command to adjust a certain time constraint. This simply allows the user to adjust the crawls depth if needed. Skipfish identifies vulnerabilities in a system as high, medium, and low priorities, based on severity.

```
(jason@jhodge)-[~]
$ sudo apt install skipfish
[sudo] password for jason:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
skipfish is already the newest version (2.10b-2kali7).
skipfish set to manually installed.
The following package was automatically installed and is no longer required:
python3-texttable
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 1378 not upgraded.
```

Here we can see the passive Kali Linux tool of my choosing, Skipfish being installed on the machine using the command “sudo apt install skipfish”.

```

(jason@jhodge)-[~]
$ skipfish -h
skipfish web application scanner - version 2.10b
Usage: skipfish [ options ... ] -W wordlist -o output_dir start_url [ start_url2 ... ]

Authentication and access options:

-A user:pass      - use specified HTTP authentication credentials
-F host=IP        - pretend that 'host' resolves to 'IP'
-C name=val       - append a custom cookie to all requests
-H name=val       - append a custom HTTP header to all requests
-b (ilf|p)        - use headers consistent with MSIE / Firefox / iPhone
-N               - do not accept any new cookies
--auth-form url   - form authentication URL
--auth-user user  - form authentication user
--auth-pass pass  - form authentication password
--auth-verify-url - URL for in-session detection

Crawl scope options:

-d max_depth      - maximum crawl tree depth (16)
-c max_child      - maximum children to index per node (512)
-x max_desc       - maximum descendants to index per branch (8192)
-r r_limit        - max total number of requests to send (100000000)
-p crawl%         - node and link crawl probability (100%)
-q hex            - repeat probabilistic scan with given seed
-I string         - only follow URLs matching 'string'
-X string         - exclude URLs matching 'string'
-K string         - do not fuzz parameters named 'string'
-D domain         - crawl cross-site links to another domain
-B domain         - trust, but do not crawl, another domain
-Z               - do not descend into 5xx locations
-O               - do not submit any forms
-P               - do not parse HTML, etc, to find new links

Reporting options:

-o dir            - write output to specified directory (required)
-M               - log warnings about mixed content / non-SSL passwords
-E               - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
-U               - log all external URLs and e-mails seen
-Q               - completely suppress duplicate nodes in reports
-u               - be quiet, disable realtime progress stats
-v               - enable runtime logging (to stderr)

Dictionary management options:

-W wordlist       - use a specified read-write wordlist (required)
-S wordlist       - load a supplemental read-only wordlist
-L               - do not auto-learn new keywords for the site
-Y               - do not fuzz extensions in directory brute-force
-R age            - purge words hit more than 'age' scans ago
-T name=val       - add new form auto-fill rule
-G max_guess      - maximum number of keyword guesses to keep (256)

-z sigfile        - load signatures from this file

Performance settings:

-g max_conn       - max simultaneous TCP connections, global (40)
-m host_conn      - max simultaneous connections, per target IP (10)
-f max_fail       - max number of consecutive HTTP errors (100)
-t req_tmout      - total request response timeout (20 s)

```

The help command, “-h” shows the available commands that could be utilized in a skipfish scan.

```
skipfish version 2.10b by lcamtuf@google.com
- www.marist.edu -

Scan statistics:
  Scan time : 1:41:36.716
  HTTP requests : 172054 (28.3/s), 3132619 kB in, 111291 kB out (532.1 kB/s)
  Compression : 2939465 kB in, 16116668 kB out (69.1% gain)
  HTTP faults : 369 net errors, 825 proto errors, 326 retried, 0 drops
  TCP handshakes : 2797 total (62.3 req/conn)
  TCP faults : 0 failures, 0 timeouts, 1 purged
  External links : 49565 skipped
  Reqs pending : 2063

Database statistics:
  Pivots : 1751 total, 391 done (22.33%)
  In progress : 891 pending, 99 init, 357 attacks, 13 dict
  Missing nodes : 487 spotted
  Node types : 1 serv, 707 dir, 132 file, 20 pinfo, 431 unkn, 437 par, 23 val
  Issues found : 78 info, 553 warn, 111 low, 817 medium, 4 high impact
  Dict size : 917 words (917 new), 8 extensions, 256 candidates
  Signatures : 77 total

[!] Scan aborted by user, bailing out!
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 1751
[+] Looking for duplicate entries: 1751
[+] Counting unique nodes: 1240
[+] Saving pivot data for third-party tools ...
[+] Writing scan description ...
[+] Writing crawl tree: 1751
[+] Generating summary views ...
[+] Report saved to 'scan/index.html' [0x39f52ba1].
[+] This was a great day for science!
```

This shows the brief results of the skipfish test I ran. In order to gain a substantial amount of data I let this test run for quite a long time and eventually ended the test. As I stated above, I chose not to limit the crawl search depth time in an effort to give the search time to find those high impact potential threats. As you can see, I discovered 4 high impact, 817 medium, 111 low, and 78 information issues.



Crawl results - click to expand:

<https://www.marist.edu/> 4 557 67 72 56 1238
Code: 200, length: 201169, declared: text/html, detected: application/xhtml+xml, charset: UTF-8 [show trace +]

Document type overview - click to expand:

application/xhtml+xml (13)
 image/svg+xml (1)
 text/html (5)

Issue type overview - click to expand:

- Query injection vector** (1)
 1. https://www.marist.edu/control_menu/?browserId=other&minifierType=&languageId=en_US&t=1697262784790 [show trace +]
Memo: response to "" different than to ""
- Server-side XML injection vector** (3)
 1. [>"></sfish></sfish><minifierType=&languageId=en_US&t=1697262784790](https://www.marist.edu/control_menu/?browserId=sfish) [show trace +]
Memo: responses for <sfish></sfish> and </sfish><sfish> look different
 2. [>"></sfish></sfish>](https://www.marist.edu/control_menu/?browserId=other&minifierType=&languageId=en_US&t=sfish) [show trace +]
Memo: responses for <sfish></sfish> and </sfish><sfish> look different
 3. [>"></sfish></sfish>](https://www.marist.edu/liberal-arts/sidebar_panel/?browserId=other&minifierType=&languageId=en_US&t=sfish) [show trace +]
Memo: responses for <sfish></sfish> and </sfish><sfish> look different
- External content embedded on a page (higher risk)** (403)
- XSS vector in document body** (154)
- HTML form with no apparent XSRF protection** (32)
- External content embedded on a page (lower risk)** (35)
- Node should be a directory, detection error?** (5)
- Response varies randomly, skipping checks** (14)
- Resource fetch failed** (53)
- Incorrect or missing MIME type (low risk)** (1)
- HTML form (not classified otherwise)** (32)
- Hidden files / directories** (8)
- New 404 signature seen** (8)
- New 'X-*' header value seen** (6)
- New 'Server' header value seen** (3)
- New HTTP cookie added** (5)
- SSL certificate issuer information** (1)

NOTE: 100 samples maximum per issue or document type.

Most of the results of this scan are all relatively low risk. That being said, there are a couple vulnerable areas for SQL injection and XML or XXE injection attacks, which can potentially be dangerous. SQL injection attacks are where an attacker can query their way into a part of a database through an application page, if this page is a login page it can be even more dangerous as they can get their hands on login information and user records. An XML or XXE injection attack is where an attacker can “interfere with the way an application processes XML data. Successful exploitation allows an attacker to view files from the application’s server and interact with any external or backend systems that the application can access” (OneHackMan). This would be something the Marist cybersecurity team can look into and make sure the system is well protected.

Wafw00f (WAF): A passive reconnaissance web application firewall (WAF) detector tool that sends HTTP requests to find out which WAF is protecting a domain.

```
(jason@jhodge)-[~]
$ sudo apt install wafw00f
[sudo] password for jason:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wafw00f is already the newest version (2.2.0-1).
wafw00f set to manually installed.
The following package was automatically installed and is no longer required:
  python3-texttable
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 1377 not upgraded.
```

Here we can see another passive Kali Linux tool of my choosing, Wafw00f being installed on the machine using the command “sudo apt install wafw00f”.

```
(jason@jhodge)-[~]
$ wafw00f -h
Usage: wafw00f url1 [url2 [url3 ... ]]
example: wafw00f http://www.victim.org/

Options:
-h, --help                show this help message and exit
-v, --verbose             Enable verbosity, multiple -v options increase
                           verbosity
-a, --findall             Find all WAFs which match the signatures, do not stop
                           testing on the first one
-r, --noredirect          Do not follow redirections given by 3xx responses
-t TEST, --test=TEST      Test for one specific WAF
-o OUTPUT, --output=OUTPUT
                           Write output to csv, json or text file depending on
                           file extension. For stdout, specify - as filename.
-f FORMAT, --format=FORMAT
                           Force output format to csv, json or text.
-i INPUT, --input-file=INPUT
                           Read targets from a file. Input format can be csv,
                           json or text. For csv and json, a `url` column name or
                           element is required.
-l, --list                List all WAFs that WAFW00F is able to detect
-p PROXY, --proxy=PROXY  Use an HTTP proxy to perform requests, examples:
                           http://hostname:8080, socks5://hostname:1080,
                           http://user:pass@hostname:8080
-V, --version             Print out the current version of WafW00f and exit.
-H HEADERS, --headers=HEADERS
                           Pass custom headers via a text file to overwrite the
                           default header set.
```

The help command, “-h” shows the available commands that could be utilized in a wafw00f test scan.

References:

AS6124 Marist College details. IPinfo. (n.d.). <https://ipinfo.io/AS6124>

Free Whois Lookup. whois.com. (n.d.). <https://www.whois.com/whois/>

How hackers attack subdomains and how to protect them. HackerNoon. (n.d.).
<https://hackernoon.com/how-hackers-attack-subdomains-and-how-to-protect-them-rc7j37f2>

IP addresses 148.100.131.0 to 148.100.131.255. IP Geolocation API and database. (n.d.-a).
<https://db-ip.com/all/148.100.131>

IP addresses 148.100.172.0 to 148.100.172.255. IP Geolocation API and database. (n.d.-b).
<https://db-ip.com/all/148.100.172>

OneHackMan. (2020, January 4). *Exploiting XML external entity (XXE) injections.* Medium.
<https://medium.com/@onehackman/exploiting-xml-external-entity-xxe-injections-b0e3eac388f9>

OSINT framework. OSINT Framework. (n.d.). <https://osintframework.com/>

Recon-NG: Kali linux tools. Kali Linux. (2022a, November 16). <https://www.kali.org/tools/recon-ng/>

WAFW00F: Kali linux tools. Kali Linux. (2022b, August 5). <https://www.kali.org/tools/wafw00f/>

Yasar, K., & Lutkevich, B. (2023, April 19). *What is a firewall and why do I need one?: Definition from TechTarget.* Security.
<https://www.techtarget.com/searchsecurity/definition/firewall>

YouTube. (2019, April 19). *Use shodan to look for vulnerable targets in a domain: Passive recon.* YouTube. <https://www.youtube.com/watch?v=Vn-Op9JEGBY>