Use Microsoft Free Security Tools as a Ninja

DEF CON 24
4 August
Las Vegas, USA



VULNEX

ME & VULNEX

Simon Roses Femerling

- Founder & CEO, VULNEX www.vulnex.com
- @simonroses
- Former Microsoft, PwC, @Stake
- US DARPA award to research on software security
- Speaker: Black Hat, RSA, HITB, OWASP, SOURCE, AppSec, DeepSec, TECHNET
- Blog: http://www.simonroses.com/
- Youtube: https://www.youtube.com/channel/UC8KUXxTSEdWfpFzAydjEzyQ

VULNEX

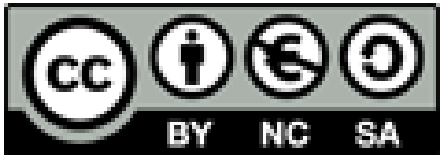
- CyberSecurity Startup
- @vulnexsl
- Professional Services & Training
- Products: BinSecSweeper (Unified File Security Analysis)

VULNEX

DISCLAIMER & LICENSE

- All Tools and resources are property of Microsoft and their authors
- Non-affiliated with Microsoft





VULNEX

WORKSHOP OBJECTIVES

- What has Microsoft to offer?
- How to improve our security posture for free!
- Development and IT Security

VULNEX

AGENDA

1. Introduction
2. Secure Development
3. IT Security
4. Conclusions

VULNEX

1. Introduction

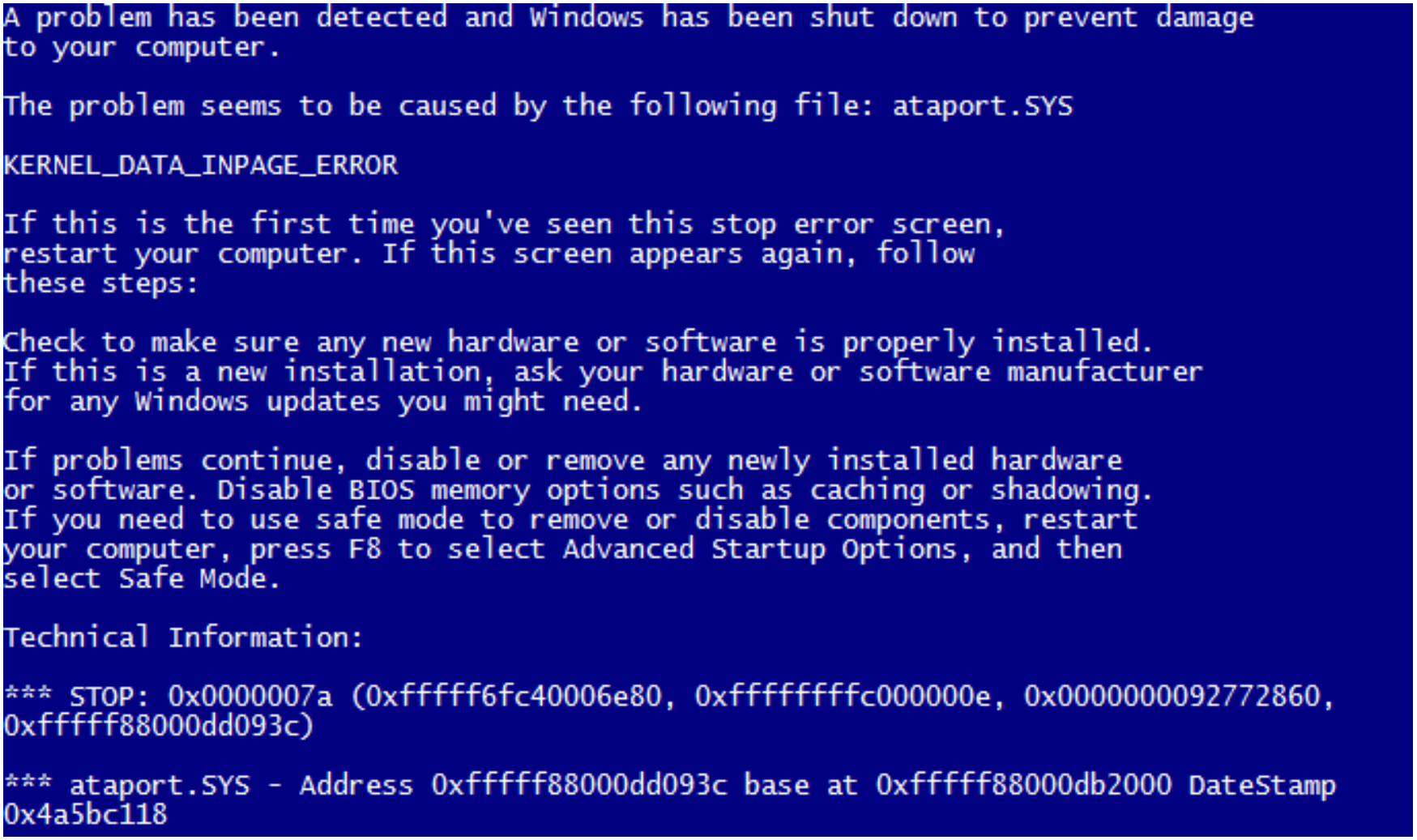
VULNEX

1. DEVELOPERS VS SYSADMINS VS ALL...



VULNEX

1. FATAL ERROR



VULNEX

1. DEFENSE IN DEPTH



VULNEX

1. MEMO FROM BILL GATES

- https://news.microsoft.com/2012/01/11/memo-from-bill-gates/#sm.001he6hz618bod7bz7k10g0w76fr0

In the past, we've made our software and services more compelling for users by adding new features and functionality, and by making our platform richly extensible. We've done a terrific job at that, but all those great features won't matter unless customers trust our software. So now, when we face a choice between adding features

VULNEX

1. MICROSOFT SDL

- The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost

- https://www.microsoft.com/en-us/SDL

1. MICROSOFT SDL



VULNEX

1. SDL: TRAINING

- **SDL Practice #1: Core Security Training**

  This practice is a prerequisite for implementing the SDL. Foundational concepts for building better software include secure design, threat modeling, secure coding, security testing, and best practices surrounding privacy

1. SDL: REQUIREMENTS

- **SDL Practice #2: Establish Security and Privacy Requirements**

  Defining and integrating security and privacy requirements early helps make it easier to identify key milestones and deliverables and minimize disruptions to plans and schedules.

- **SDL Practice #3: Create Quality Gates/Bug Bars**

  Defining minimum acceptable levels of security and privacy quality at the start helps a team understand risks associated with security issues, identify and fix security bugs during development, and apply the standards throughout the entire project.

- **SDL Practice #4: Perform Security and Privacy Risk Assessments**

  Examining software design based on costs and regulatory requirements helps a team identify which portions of a project will require threat modeling and security design reviews before release and determine the Privacy Impact Rating of a feature, product, or service.

VULNEX

1. SDL: DESIGN

- **SDL Practice #5: Establish Design Requirements**

  Considering security and privacy concerns early helps minimize the risk of schedule disruptions and reduce a project's expense.

- **SDL Practice #6: Attack Surface Analysis/Reduction**

  Reducing the opportunities for attackers to exploit a potential weak spot or vulnerability requires thoroughly analyzing overall attack surface and includes disabling or restricting access to system services, applying the principle of least privilege, and employing layered defenses wherever possible.

- **SDL Practice #7: Use Threat Modeling**

  Applying a structured approach to threat scenarios during design helps a team more effectively and less expensively identify security vulnerabilities, determine risks from those threats, and establish appropriate mitigations.

1. SDL: IMPLEMENTATION

- **SDL Practice #8: Use Approved Tools**

  Publishing a list of approved tools and associated security checks (such as compiler/linker options and warnings) helps automate and enforce security practices easily at a low cost. Keeping the list regularly updated means the latest tool versions are used and allows inclusion of new security analysis functionality and protections.

- **SDL Practice #9: Deprecate Unsafe Functions**

  Analyzing all project functions and APIs and banning those determined to be unsafe helps reduce potential security bugs with very little engineering cost. Specific actions include using header files, newer compilers, or code scanning tools to check code for functions on the banned list, and then replacing them with safer alternatives.

- **SDL Practice #10: Perform Static Analysis**

  Analyzing the source code prior to compile provides a scalable method of security code review and helps ensure that secure coding policies are being followed.

1. SDL: VERIFICATION

- **SDL Practice #11: Perform Dynamic Analysis**

  Performing run-time verification checks software functionality using tools that monitor application behavior for memory corruption, user privilege issues, and other critical security problems.

- **SDL Practice #12: Fuzz Testing**

  Inducing program failure by deliberately introducing malformed or random data to an application helps reveal potential security issues prior to release while requiring modest resource investment.

- **SDL Practice #13: Attack Surface Review**

  Reviewing attack surface measurement upon code completion helps ensure that any design or implementation changes to an application or system have been taken into account, and that any new attack vectors created as a result of the changes have been reviewed and mitigated including threat models.

VULNEX

1. SDL: RELEASE

- **SDL Practice #14: Create an Incident Response Plan**

  Preparing an Incident Response Plan is crucial for helping to address new threats that can emerge over time. It includes identifying appropriate security emergency contacts and establishing security servicing plans for code inherited from other groups within the organization and for licensed third-party code.

- **SDL Practice #15: Conduct Final Security Review**

  Deliberately reviewing all security activities that were performed helps ensure software release readiness. The Final Security Review (FSR) usually includes examining threat models, tools outputs, and performance against the quality gates and bug bars defined during the Requirements Phase.

- **SDL Practice #16: Certify Release and Archive**

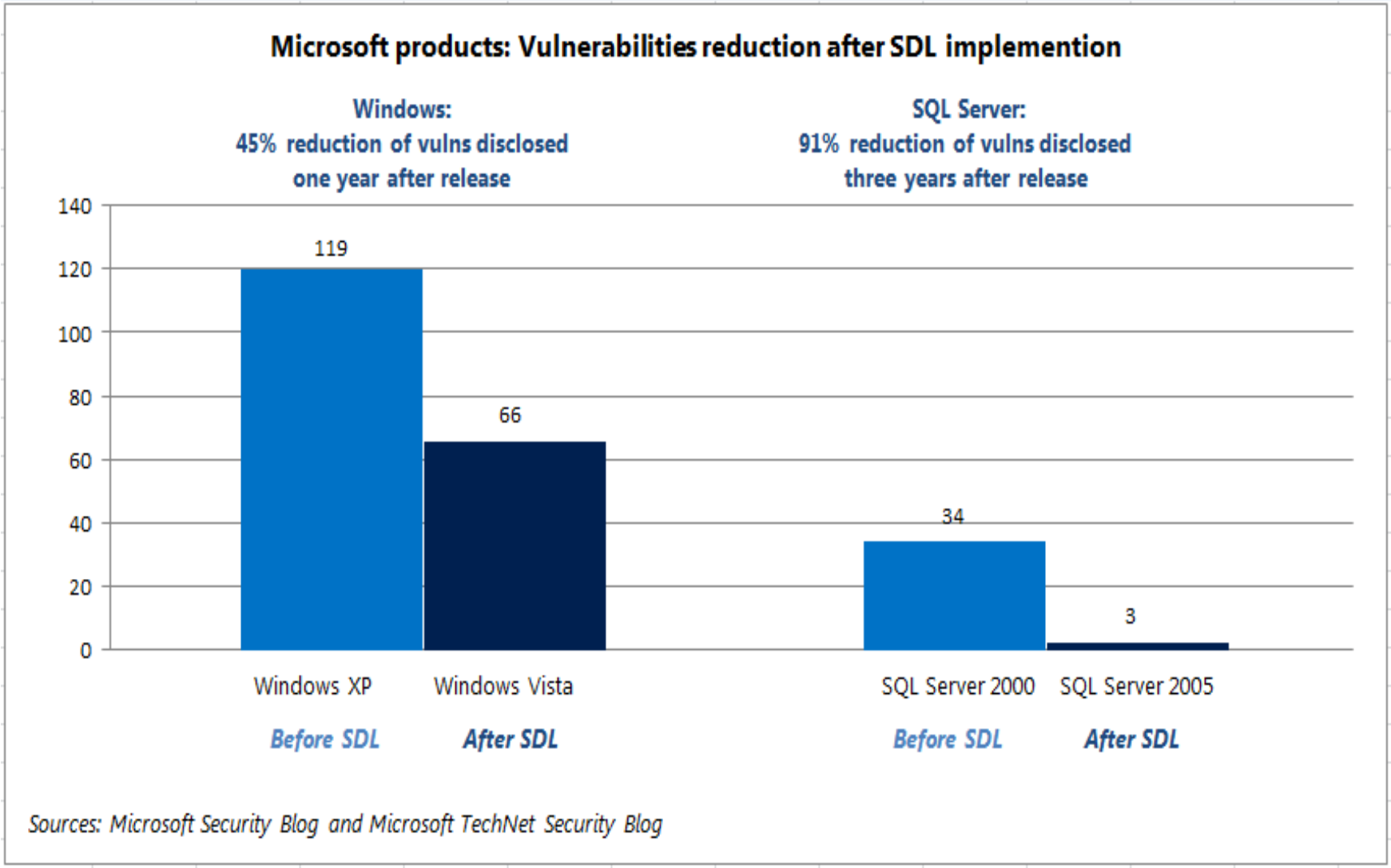
  Certifying software prior to a release helps ensure security and privacy requirements were met. Archiving all pertinent data is essential for performing post-release servicing tasks and helps lower the long-term costs associated with sustained software engineering.

1. SDL: RESPONSE

- **SDL Practice #17: Execute Incident Response Plan**

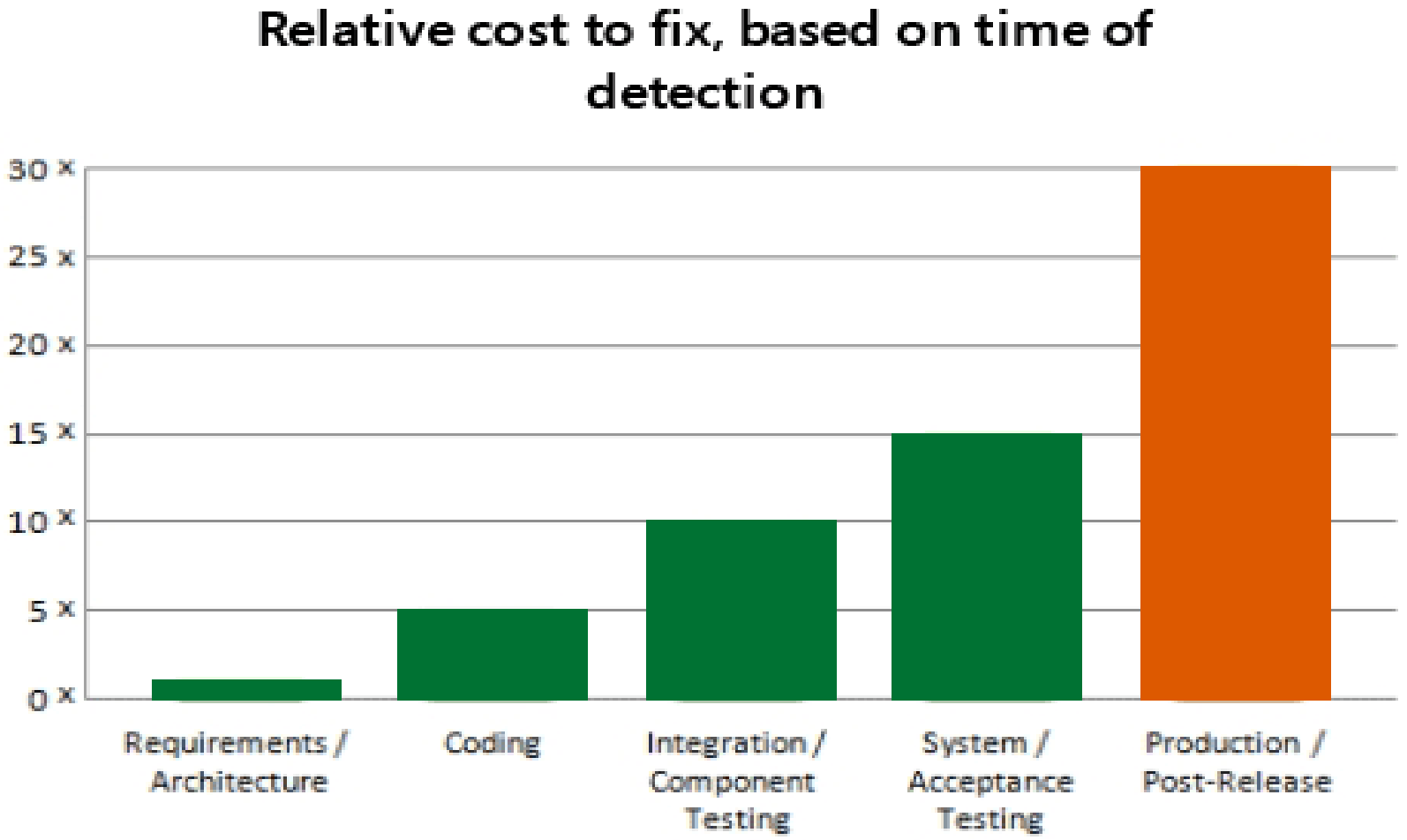
  Being able to implement the Incident Response Plan instituted in the Release phase is essential to helping protect customers from software security or privacy vulnerabilities that emerge.

1. REDUCING VULNERABILITIES



VULNEX

1. REDUCING COSTS



VULNEX

1. SYSINTERNALS

- Not about Sysinternals suite
- Awesome tools!
- https://technet.microsoft.com/en-us/sysinternals/bb545021

2. Secure Development

VULNEX

2. AVAILABLE SECURE DEVELOPMENT TOOLS

1. Microsoft Solutions Framework (MSF) for Capability Maturity Model Integration (CMMI) 2013 plus Security Development Lifecycle (SDL)
2. Microsoft Solutions Framework (MSF) for Agile 2013 plus Security Development Lifecycle (SDL)
3. TM SDL 2016
4. AntiXSS
5. Visual Studio 2012 / 2015
6. FXCOP
7. CAT.NET
8. SDL REGEX FUZZER
9. SDL MINIFUZZ
10. App Verifier
11. BinScope
12. Binskim

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI) 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)

- Version: 1.0

- Downloadable template that integrates the Microsoft Security Development Lifecycle (SDL) directly into your Visual Studio Team Foundation Server 2013 software development environment.

- Requires Visual Studio Team Foundation Server 2013

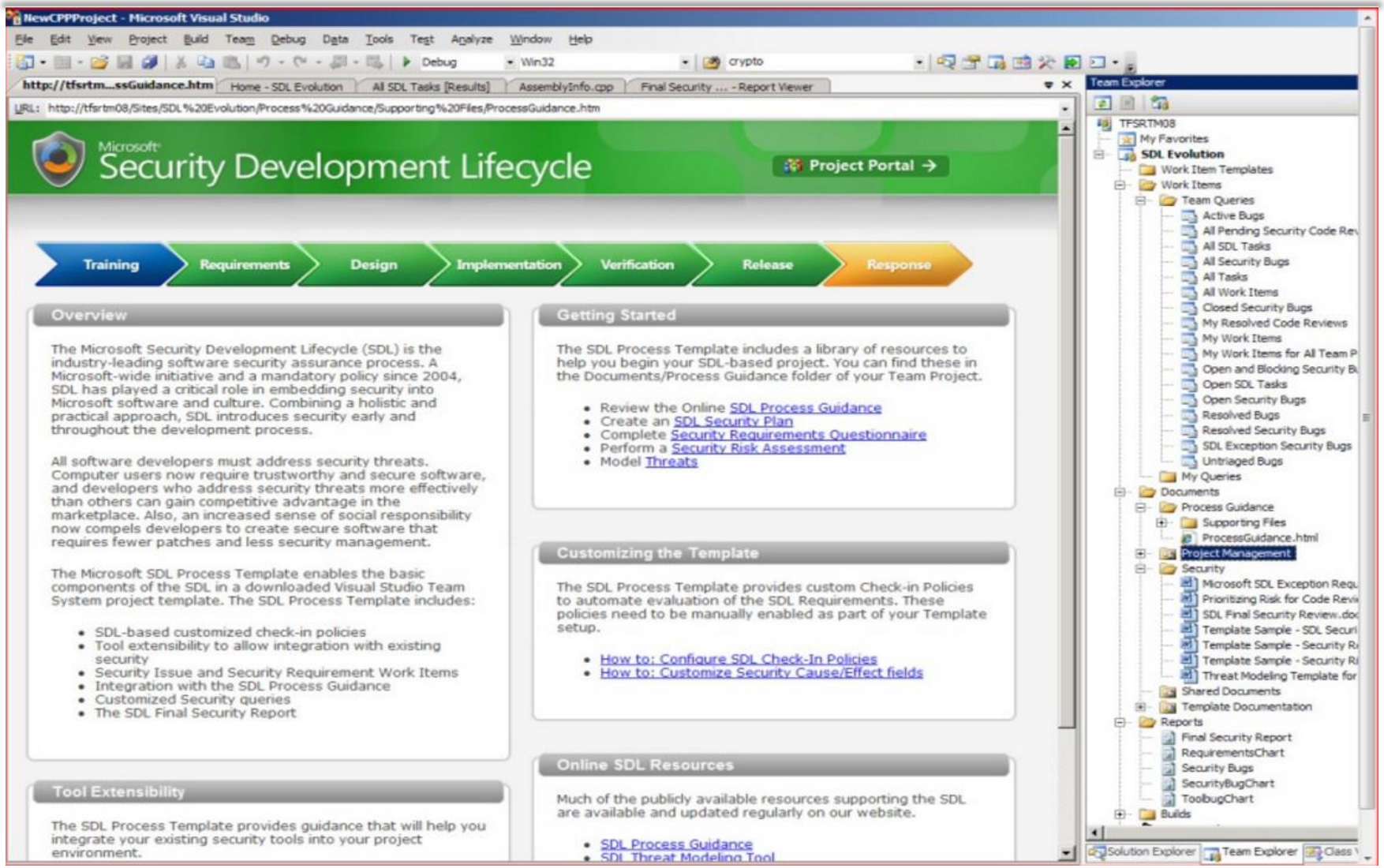
- More info: https://www.microsoft.com/en-us/SDL/adopt/processtemplate.aspx
  Download: https://www.microsoft.com/en-us/download/details.aspx?id=42519

VULNEX

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI) 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)

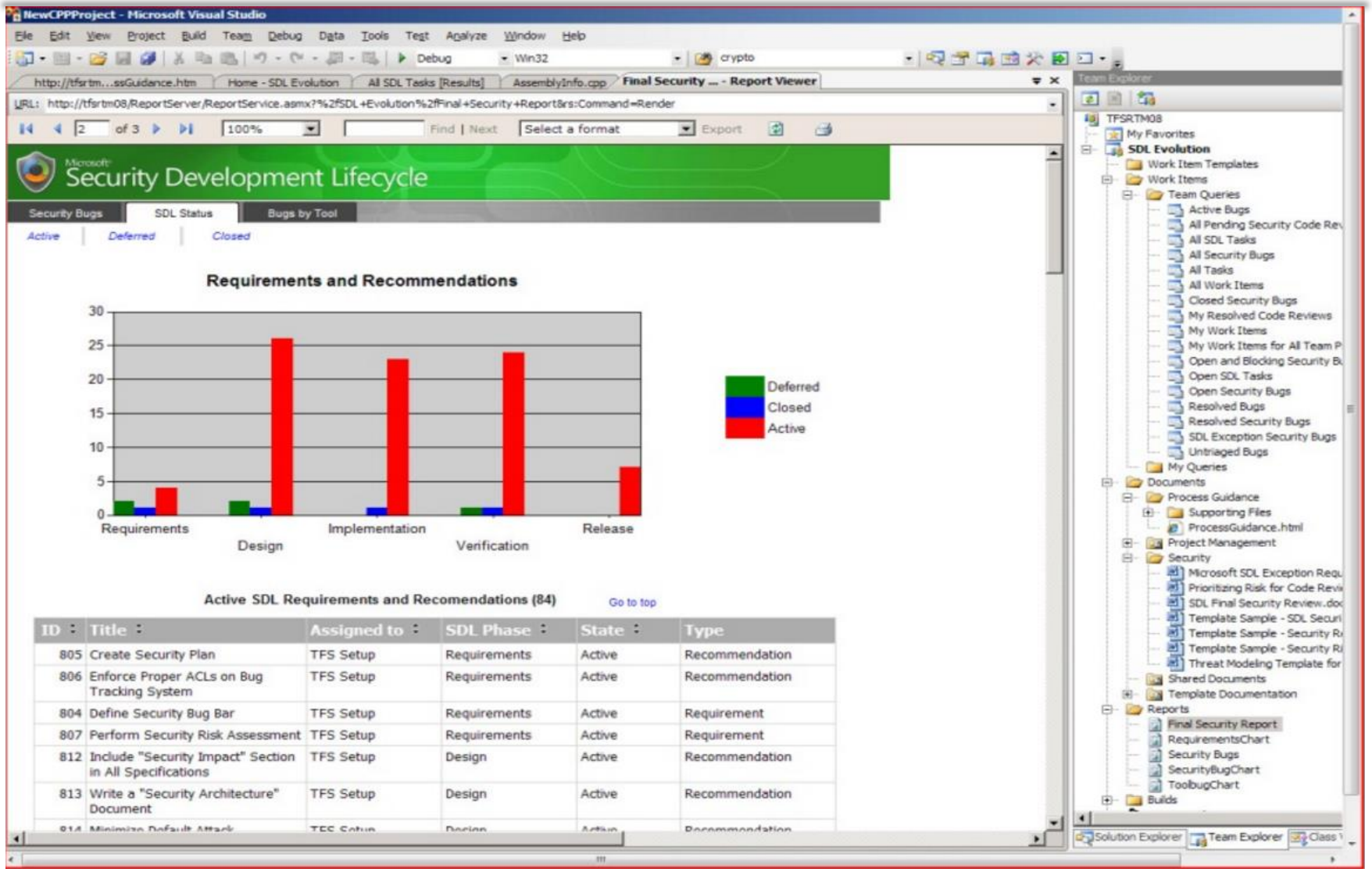
FEATURES
SDL requirements
SDL policies
Custom vulnerabilities queries
SDL guides & resources
Final Security Review (FSR) report
Third party tool integration
Security templates

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI) 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)



VULNEX

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR CAPABILITY MATURITY MODEL INTEGRATION (CMMI) 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)

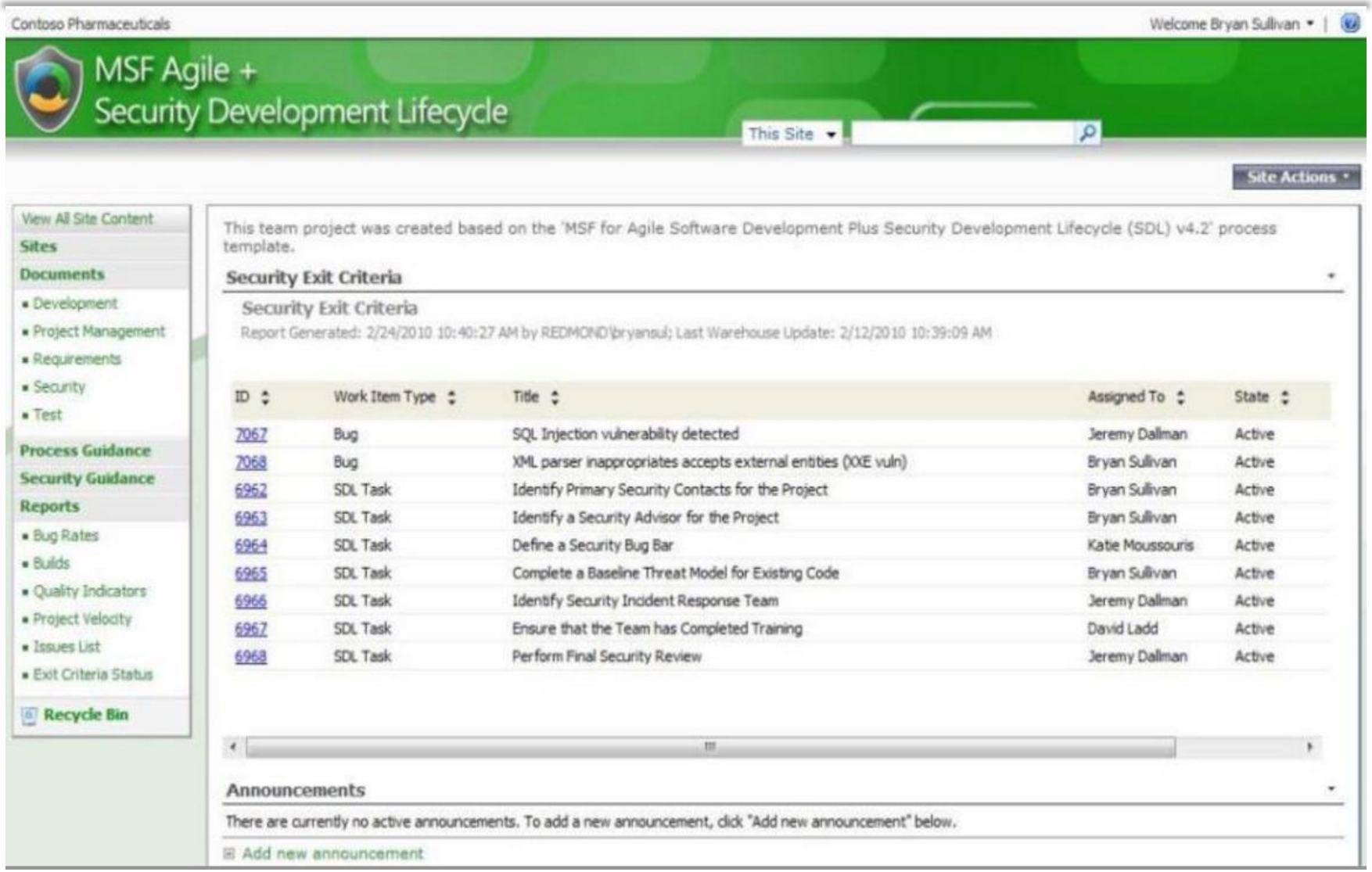


VULNEX

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR AGILE 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)

- Version: 1.0
- Same as before but for Agile development
- Requires Visual Studio Team Foundation Server 2013
- More info: https://www.microsoft.com/en-us/SDL/adopt/agile.aspx
  Download: https://www.microsoft.com/en-us/download/details.aspx?id=42517

2. TOOL: MICROSOFT SOLUTIONS FRAMEWORK (MSF) FOR AGILE 2013 PLUS SECURITY DEVELOPMENT LIFECYCLE (SDL)



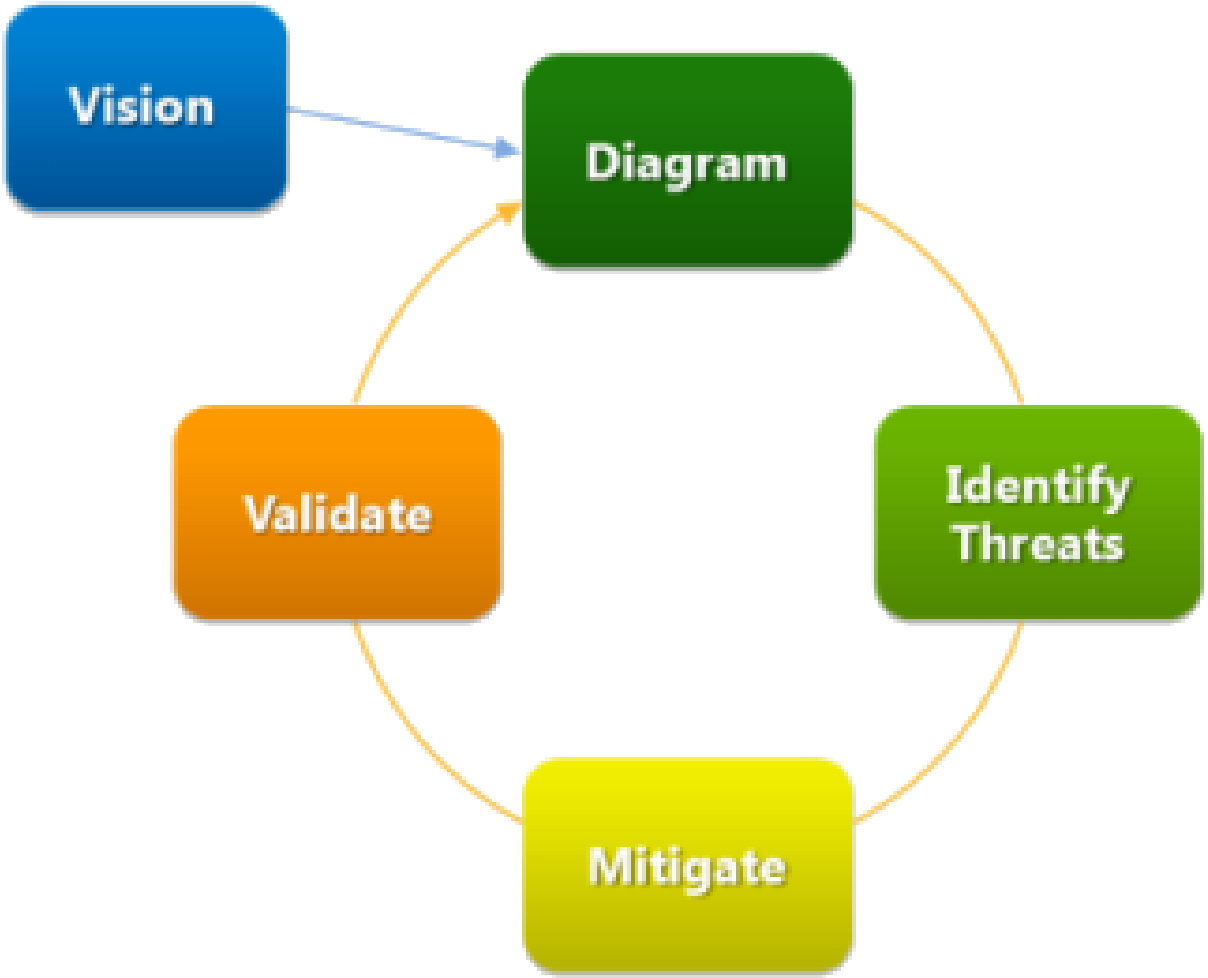
VULNEX

2. TOOL: SDL TM 2016

- Version: 2016
- Threat Modeling
- Find threats during design phase, determine threats and define appropriate mitigations and distribute security tasks across stakeholders
- More info: https://blogs.microsoft.com/cybertrust/2015/10/07/whats-new-with-microsoft-threat-modeling-tool-2016/
  Download: https://www.microsoft.com/en-us/download/details.aspx?id=49168

VULNEX

2. TOOL: SDL TM 2016



VULNEX

2. TOOL: SDL TM 2016

STRIDE
Spoofing
Tampering
Repudiation
Information **D**isclosure
Elevation of Privilege

VULNEX

2. TOOL: SDL TM 2016



VULNEX

2. TOOL: BANNED.H

- Version: 2.0
- Insecure functions banned by the SDL
- Visual Studio replaces them under the hood by a more secure version
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=24817

VULNEX

2. TOOL: BANNED.H

banned.h

Miscellaneous Files | (Global Scope)

```
        // SDL 5.0 and later Requirements
#       if defined(_STRSAFE_H_INCLUDED_) && !defined(STRSAFE_NO_DEPRECATE)

            // Only deprecate what's not already deprecated by StrSafe
#           pragma deprecated (_mbscpy, _mbccpy)
#           pragma deprecated (strcatA, strcatW, _mbscat, StrCatBuff, StrCatBuffA, StrCatBuffW, StrCatChainW, _tccat, _mbccat)
#           pragma deprecated (strncpy, wcsncpy, _tcsncpy, _mbsncpy, _mbsnbcpy, StrCpyN, StrCpyNA, StrCpyNW, StrNCpy, strcpynA, StrNCpyA, StrNCpyW, lstrcpyn, lstrcpynA, lstrcpynW)
#           pragma deprecated (strncat, wcsncat, _tcsncat, _mbsncat, _mbsnbcat, lstrncat, lstrcatnA, lstrcatnW, lstrcatn)
#           pragma deprecated (IsBadWritePtr, IsBadHugeWritePtr, IsBadReadPtr, IsBadHugeReadPtr, IsBadCodePtr, IsBadStringPtr)
#           pragma deprecated (memcpy, RtlCopyMemory, CopyMemory, wmemcpy)

#       else
            // StrSafe not loaded, so deprecate everything!
#           pragma deprecated (strcpy, strcpyA, strcpyW, wcscpy, _tcscpy, _mbscpy, StrCpy, StrCpyA, StrCpyW, lstrcpy, lstrcpyA, lstrcpyW, _tccpy, _mbccpy, _ftcscpy)
#           pragma deprecated (strcat, strcatA, strcatW, wcscat, _tcscat, _mbscat, StrCat, StrCatA, StrCatW, lstrcat, lstrcatA, lstrcatW, StrCatBuff, StrCatBuffA, StrCatBuffW, StrCatChainW, _tccat, _mbccat, _
#           pragma deprecated (sprintfW, sprintfA, wsprintf, wsprintfW, wsprintfA, sprintf, swprintf, _stprintf)
#           pragma deprecated (wvsprintf, wvsprintfA, wvsprintfW, vsprintf, _vstprintf, vswprintf)
#           pragma deprecated (strncpy, wcsncpy, _tcsncpy, _mbsncpy, _mbsnbcpy, StrCpyN, StrCpyNA, StrCpyNW, StrNCpy, strcpynA, StrNCpyA, StrNCpyW, lstrcpyn, lstrcpynA, lstrcpynW)
#           pragma deprecated (strncat, wcsncat, _tcsncat, _mbsncat, _mbsnbcat, StrCatN, StrCatNA, StrCatNW, StrNCat, StrNCatA, StrNCatW, lstrncat, lstrcatnA, lstrcatnW, lstrcatn)
#           pragma deprecated (gets, _getts, _gettws)
#           pragma deprecated (IsBadWritePtr, IsBadHugeWritePtr, IsBadReadPtr, IsBadHugeReadPtr, IsBadCodePtr, IsBadStringPtr)
#           pragma deprecated (memcpy, RtlCopyMemory, CopyMemory, wmemcpy)
#       endif //defined(_STRSAFE_H_INCLUDED_) && !defined(STRSAFE_NO_DEPRECATE)

// SDL 5.0 and later Recommendations
#       if defined(_SDL_BANNED_RECOMMENDED)
#           if defined(_STRSAFE_H_INCLUDED_) && !defined(STRSAFE_NO_DEPRECATE)
                // Only deprecate what's not already deprecated by StrSafe
#               pragma deprecated (wnsprintf, wnsprintfA, wnsprintfW)
#               pragma deprecated (vsnprintf, wvnsprintf, wvnsprintfA, wvnsprintfW)
#               pragma deprecated (strtok, _tcstok, wcstok, _mbstok)
#               pragma deprecated (makepath, _tmakepath,  _makepath, _wmakepath)
#               pragma deprecated (_splitpath, _tsplitpath, _wsplitpath)
#               pragma deprecated (scanf, wscanf, _tscanf, sscanf, swscanf, _stscanf, snscanf, snwscanf, _sntscanf)
#               pragma deprecated (_itoa, _itow, _i64toa, _i64tow, _ui64toa, _ui64tot, _ui64tow, _ultoa, _ultot, _ultow)
#               pragma deprecated (CharToOem, CharToOemA, CharToOemW, OemToChar, OemToCharA, OemToCharW, CharToOemBuffA, CharToOemBuffW)
#               pragma deprecated (alloca, _alloca)
#               pragma deprecated (strlen, wcslen, _mbslen, _mbstrlen, StrLen, lstrlen)
#               pragma deprecated (ChangeWindowMessageFilter)
#           else
                // StrSafe not loaded, so deprecate everything!
#               pragma deprecated (wnsprintf, wnsprintfA, wnsprintfW, _snwprintf, _snprintf, _sntprintf)
#               pragma deprecated (_vsnprintf, vsnprintf, _vsnwprintf, _vsntprintf, wvnsprintf, wvnsprintfA, wvnsprintfW)
#               pragma deprecated (strtok, _tcstok, wcstok, _mbstok)
#               pragma deprecated (makepath, _tmakepath,  _makepath, _wmakepath)
#               pragma deprecated (_splitpath, _tsplitpath, _wsplitpath)
#               pragma deprecated (scanf, wscanf, _tscanf, sscanf, swscanf, _stscanf, snscanf, snwscanf, _sntscanf)
#               pragma deprecated (_itoa, _itow, _i64toa, _i64tow, _ui64toa, _ui64tot, _ui64tow, _ultoa, _ultot, _ultow)
#               pragma deprecated (CharToOem, CharToOemA, CharToOemW, OemToChar, OemToCharA, OemToCharW, CharToOemBuffA, CharToOemBuffW)
#               pragma deprecated (alloca, _alloca)
#               pragma deprecated (strlen, wcslen, _mbslen, _mbstrlen, StrLen, lstrlen)
#               pragma deprecated (ChangeWindowMessageFilter)
#           endif // StrSafe
#       endif // SDL recommended
```

VULNEX

2. TOOL: ANTIXSS

- Version: 4.3
- Library to mitigate the potential of Cross-Site Scripting (XSS) attacks in web-based applications
- AKA: Microsoft Web Protection Library
- Two components:
  - Development library
  - Security Runtime Engine (SRE) – XSS y SQLi
- Included by default starting .NET 4.0 (Standalone end of life) https://msdn.microsoft.com/en-us/library/system.web.security.antixss.antixssencoder(v=vs.110).aspx
- More info:
  https://wpl.codeplex.com/
  https://www.microsoft.com/en-us/download/details.aspx?id=28589

VULNEX

2. TOOL: ANTIXSS

Method	Description
HtmlEncode	Decodes a value from an HTML-encoded string
HtmlAtributeEncode	Encodes and outputs the specified string for use in an HTML attribute
XmlEncode	Encodes the specified string for use in XML attributes
XmlAtributeEncode	Encodes the specified string for use in XML attributes
UrlEncode	Encodes the specified string for use in a URL
UrlPathEncode	Encodes path strings for use in a URL
JavaScriptEncode	Encodes a string

2. TOOL: ANTIXSS

1. Use

1. Add a using directive; `using Microsoft.Security.Application;`
2. Change the code which assigns output, for example

   ```
   string Name = Request.QueryString["Name"];
   ```

   would become

   ```
   string Name = Encoder.HtmlEncode(Request.QueryString["Name"]);
   ```

2. Example

```
HtmlInputText input3 = new HtmlInputText();
input3.Value = AntiXss.HtmlEncode("<script>alert('hacked!')</script>");
```

VULNEX

2. TOOL : VISUAL STUDIO 2015

- Version: 2015
- Microsoft Development Environment
- More info:
  https://www.visualstudio.com
- VS Secure Documentation:
  https://msdn.microsoft.com/en-us/library/k3a3hzw7.aspx
  https://msdn.microsoft.com/en-us/library/jj161081.aspx
  https://msdn.microsoft.com/en-us/library/4cftbc6c.aspx

2. TOOL: VISUAL STUDIO 2015

VS SECURITY FLAGS	DESCRIPTION
/guard	Analyze control flow for indirect call targets at compile time
/GS	Insert overrun detection code into functions that are at risk of being exploited
/SAFESEH	Prevent the execution of exception handlers that are introduced by a malicious attack
/NXCOMPAT	DEP guards the CPU against the execution of non-code pages
/analyze	Reports potential security issues such as buffer overrun, un-initialized memory, null pointer dereferencing, and memory leaks
/DYNAMICBASE	Address Space Layout Randomisation
/SDL	Enables a superset of the baseline security checks (Compile-time & Runtime checks)

2. TOOL: VISUAL STUDIO 2015

/SDL – Compile-time checks

Warning	Command line switch	Description
C4146	/we4146	A unary minus operator was applied to an unsigned type, resulting in an unsigned result
C4308	/we4308	A negative integral constant converted to unsigned type, resulting in a possibly meaningless result
C4532	/we4532	Use of “continue”, “break” or “goto” keywords in a __finally/finally block has undefined behavior during abnormal termination
C4533	/we4533	Code initializing a variable will not be executed
C4700	/we4700	Use of an uninitialized local variable
C4789	/we4789	Buffer overrun when specific C run-time (CRT) functions are used
C4995	/we4995	Use of a function marked with pragma deprecated
C4996	/we4996	Use of a function marked as deprecated

VULNEX

2. TOOL: VISUAL STUDIO 2015

/SDL – Runtime checks
Enables the strict mode of /GS run-time buffer overrun detection
Performs limited pointer sanitization
Performs class member initialization

2. TOOL : VISUAL STUDIO 2015

- Note: Visual Studio 2015 Update 1 and 2 add telemetry function calls into binaries
- Compile from command line to remove functionality:
  - notelemetry.obj
- https://www.reddit.com/r/cpp/comments/4ibauu/visual_studio_adding_telemetry_function_calls_to/d30dmvu

2. TOOL: VISUAL STUDIO 2015



VULNEX

2. TOOL: FXCOP

- Version: 10.0
- Static code analysis for managed applications
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=8279

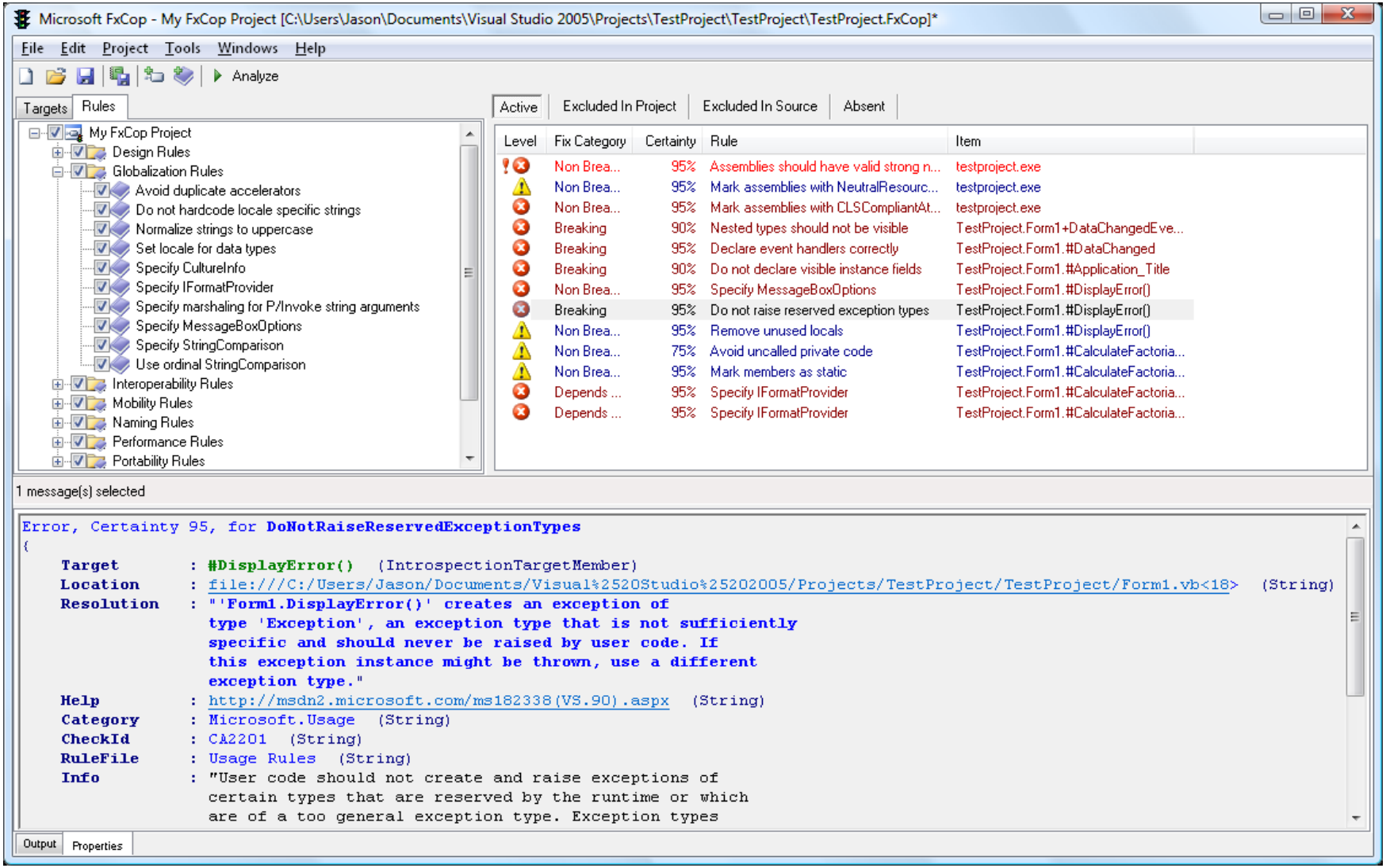
VULNEX

2. TOOL: FXCOP

FXCOP RULES
COM
Design
Globals
Names
Performances
Security
Interaction between managed and native code
.NET Code Access Security
Exposed interfaces in code
Best practices
Memory

VULNEX

2. TOOL: FXCOP



VULNEX

2. TOOL: CAT.NET

- Version: 2.0
- .NET static analysis (source code / binaries)
- GUI and Command Line
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=5570 (v1 x64)
  http://blogs.msdn.com/b/securitytools/archive/2009/11/12/how-to-run-cat-net-2-0-ctp.aspx (v2 Beta)

VULNEX

2. TOOL: CAT.NET

CAT.NET SECURITY RULES
Cross-Site Scripting (XSS)
SQL Injection
LDAP Injection
XPATH Injection
Redirections
Process Command Execution
File Canonicalization
Exception Disclosure

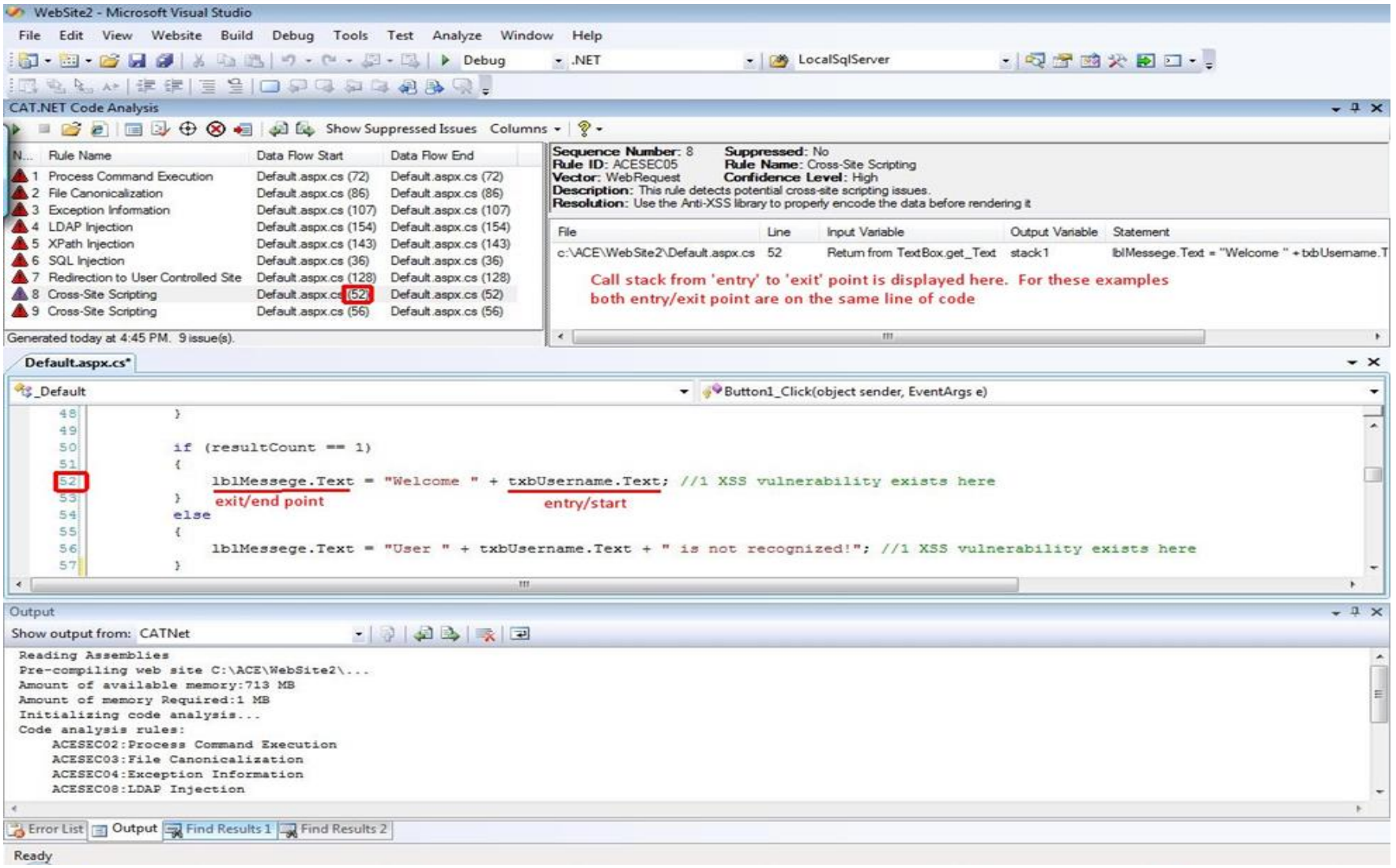
VULNEX

2. TOOL: CAT.NET

OWASP TOP 10 - 2013
A1 - Injection
A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References
A5 – Security Misconfiguration
A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control
A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards

VULNEX

2. TOOL: CAT.NET



VULNEX

2. TOOL: CAT.NET



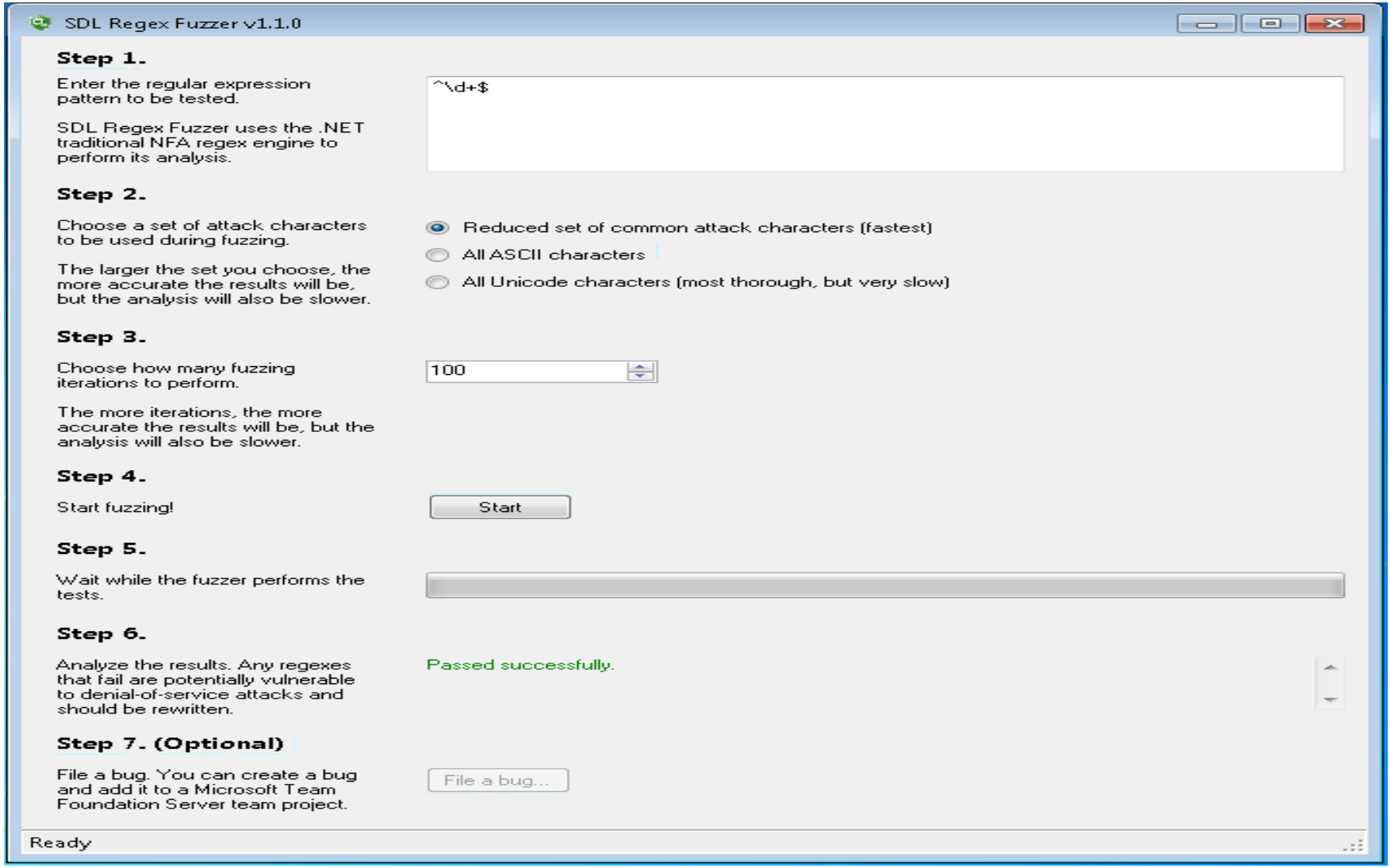
VULNEX

2. TOOL: SDL REGEX FUZZER

- Version: 1.1.0
- Regular expression (REGEX) fuzzer to identify DoS
- Download: http://www.microsoft.com/en-us/download/details.aspx?id=20095

VULNEX

2. TOOL: SDL REGEX FUZZER



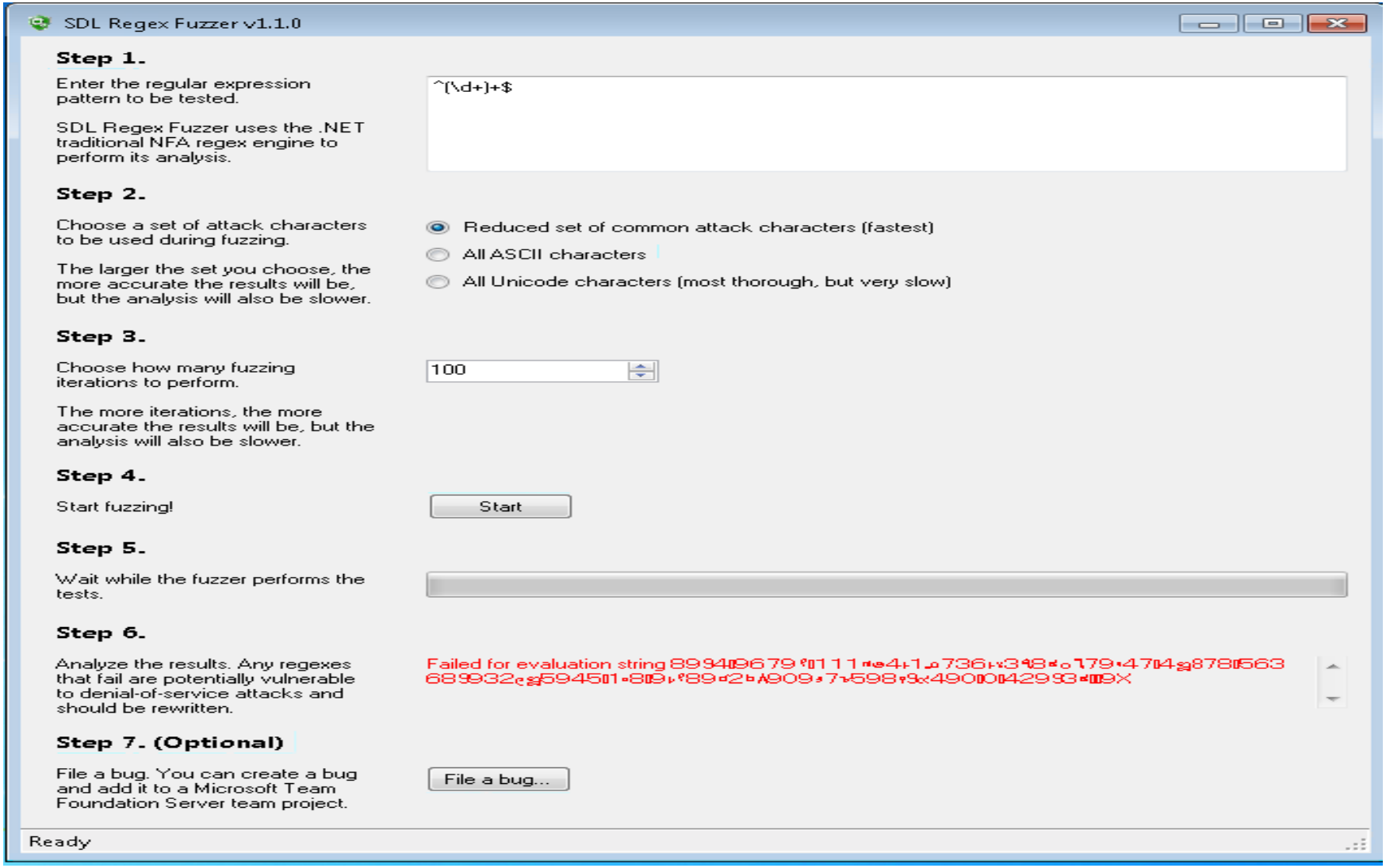
VULNEX

2. TOOL: SDL REGEX FUZZER



VULNEX

2. TOOL: SDL REGEX FUZZER



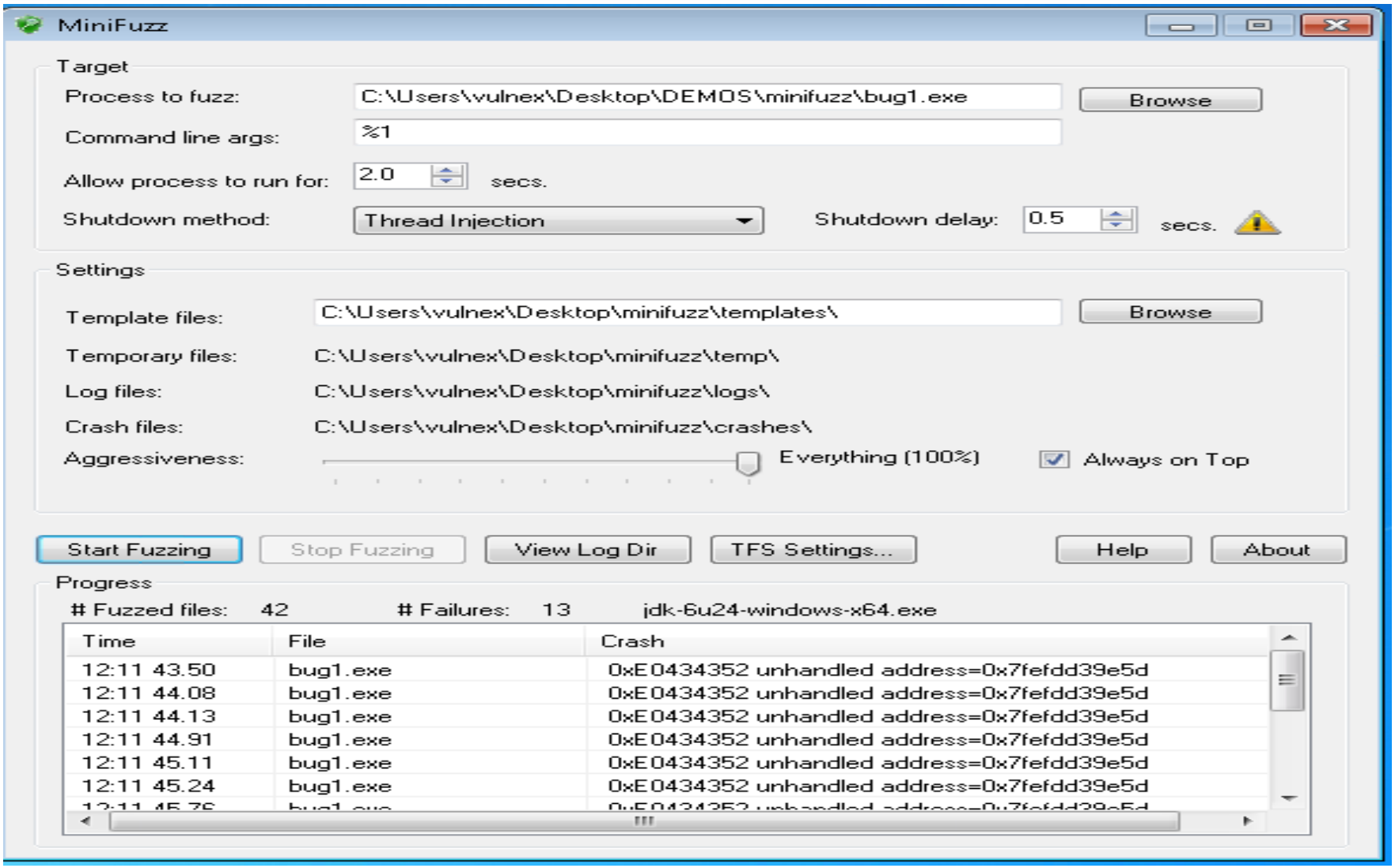
VULNEX

2. TOOL: SDL MINIFUZZ

- Version: 1.5.5.0
- Command line fuzzer
- Easy to use
- Download: www.microsoft.com/en-us/download/details.aspx?id=21769

VULNEX

2. TOOL: SDL MINIFUZZ



VULNEX

2. TOOL: SDL MINIFUZZ



VULNEX

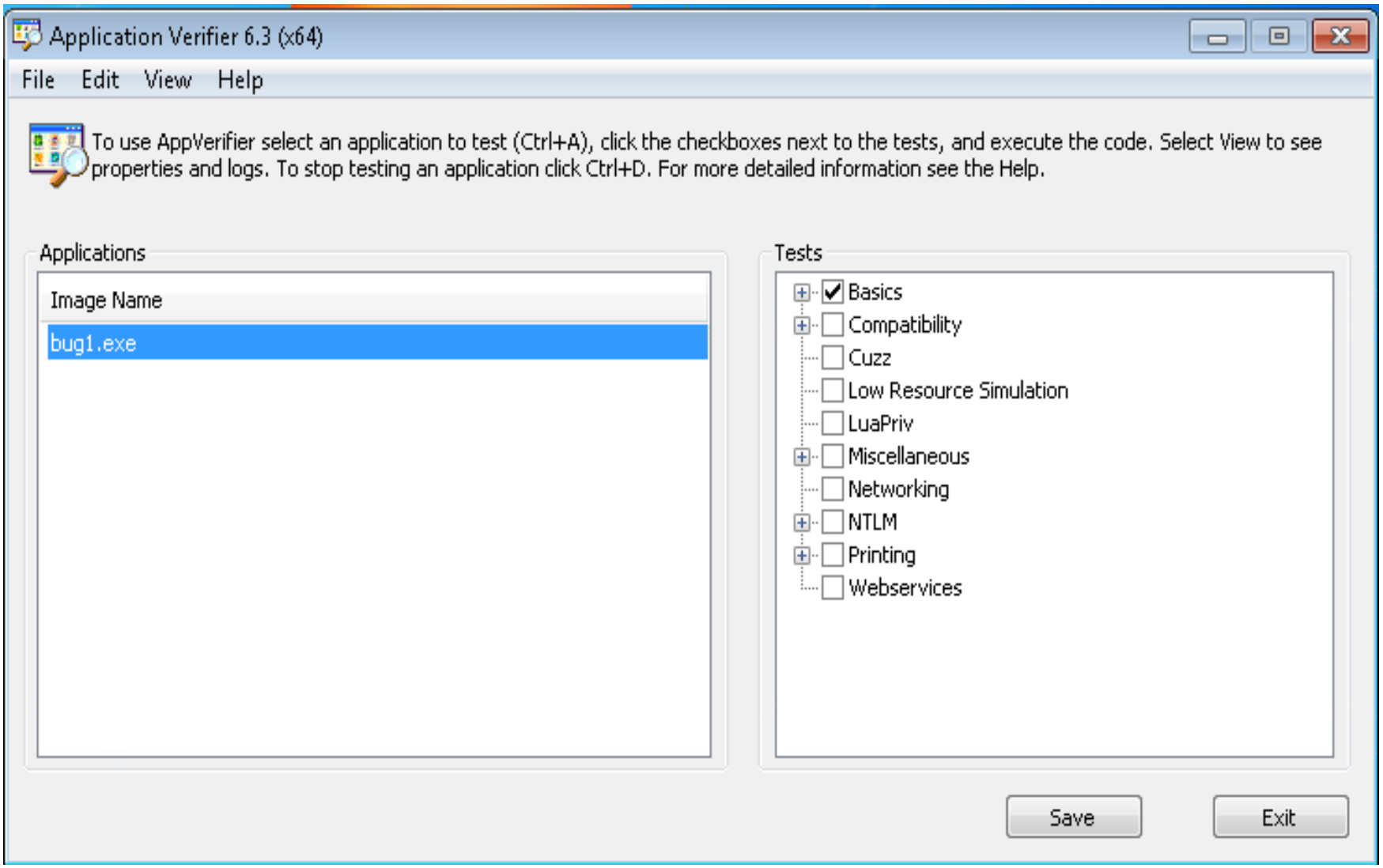
2. TOOL: APP VERIFIER

- Version: 4.0.665
- Runtime bug catcher
- Analyze C++ programs
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=20028

2. TOOL: APP VERIFIER

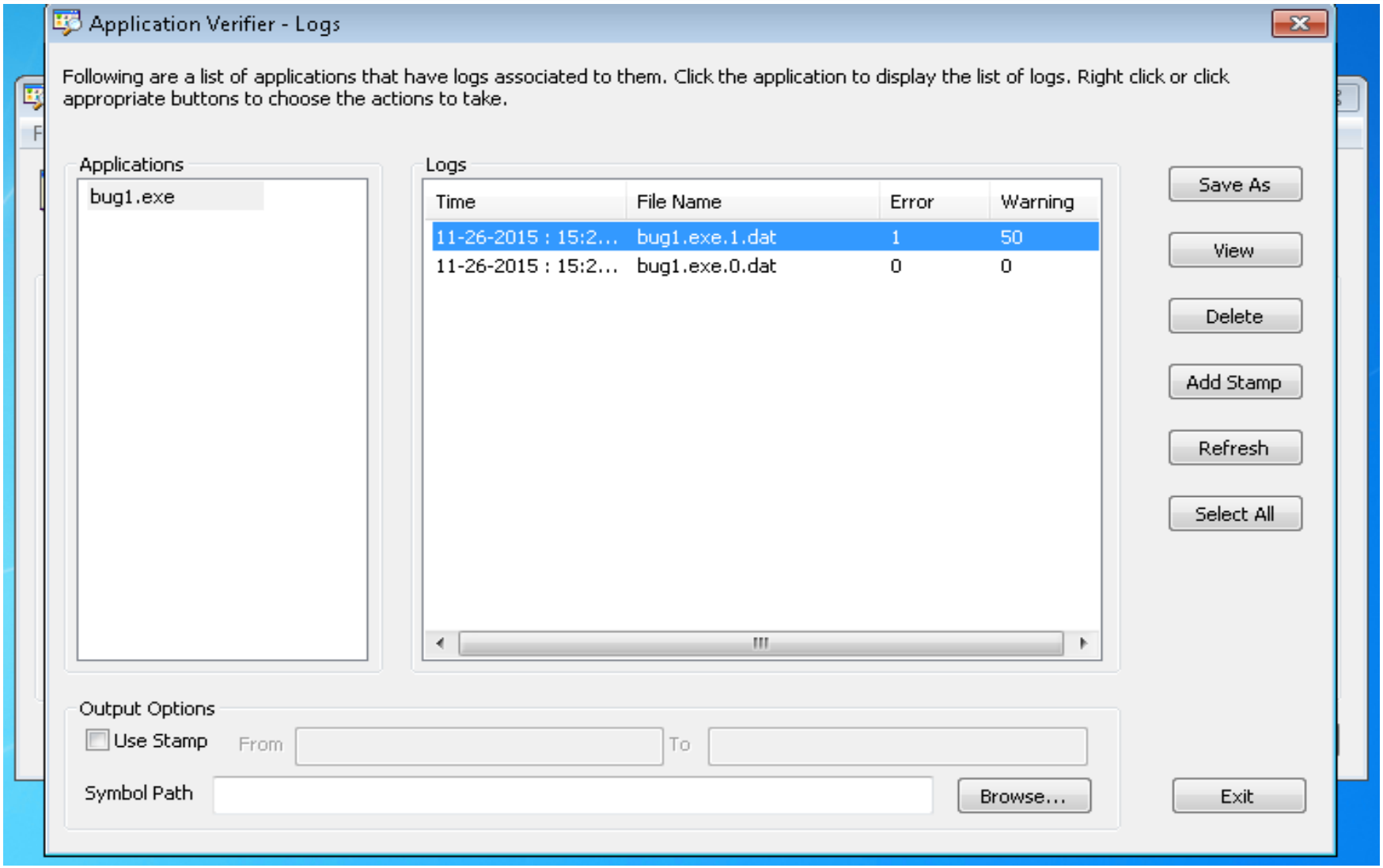
APP VERIFIER RULES
Heaps
Handles
Locks
TLS
Memory
Exceptions
Threadpool
Low Resources simulation

2. TOOL: APP VERIFIER



VULNEX

2. TOOL: APP VERIFIER



VULNEX

2. TOOL: APP VERIFIER



VULNEX

2. TOOL: BINSCOPE

- Version: 2014
- Analyzes binaries for SDL compilation best practices (Managed and native)
- Last version command line only
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=44995

VULNEX

2. TOOL: BINSCOPE

BINSCOPE RULES
Missing Build Time Flags
/GS
/SAFESEH
/NXCOMPAT
/DYNAMICBASE
Binary Features
Global function pointers
Shared read/write sections
Partially trusted called managed assemblies
Compiler version

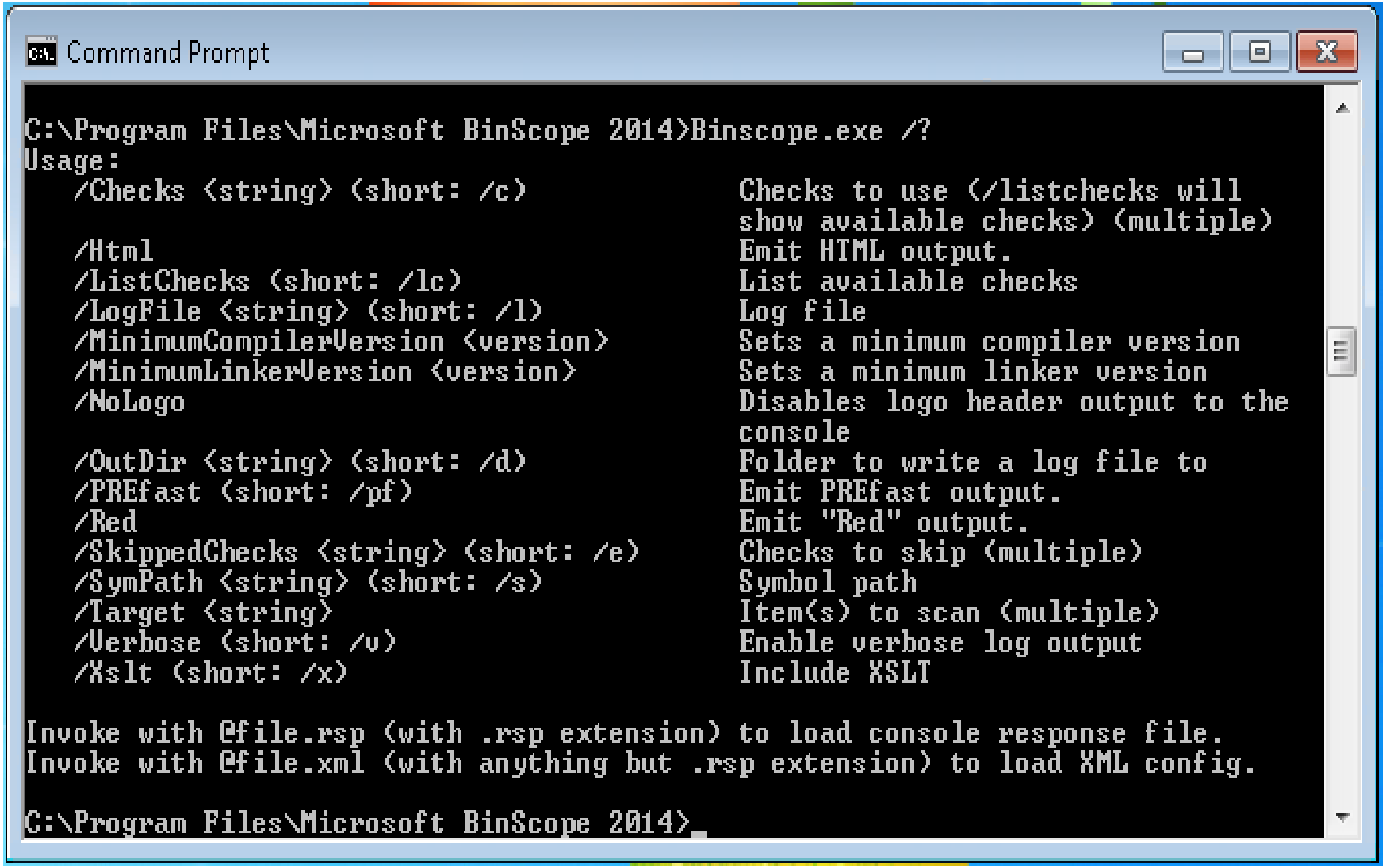
2. TOOL: BINSCOPE

BINSCOPE CHECK	SDL
AppContainerCheck *(Required for Windows Store Certification)*	NO
ATLVersionCheck	YES
ATLVulnCheck	YES
CompilerVersionCheck	YES
DBCheck	YES
DefaultGSCookieCheck	YES
ExecutableImportsCheck	YES
FunctionPointersCheck	NO
GSCheck	YES
GSFriendlyInitCheck	YES
GSFunctionSafeBuffersCheck	YES
HighEntropyVACheck	YES

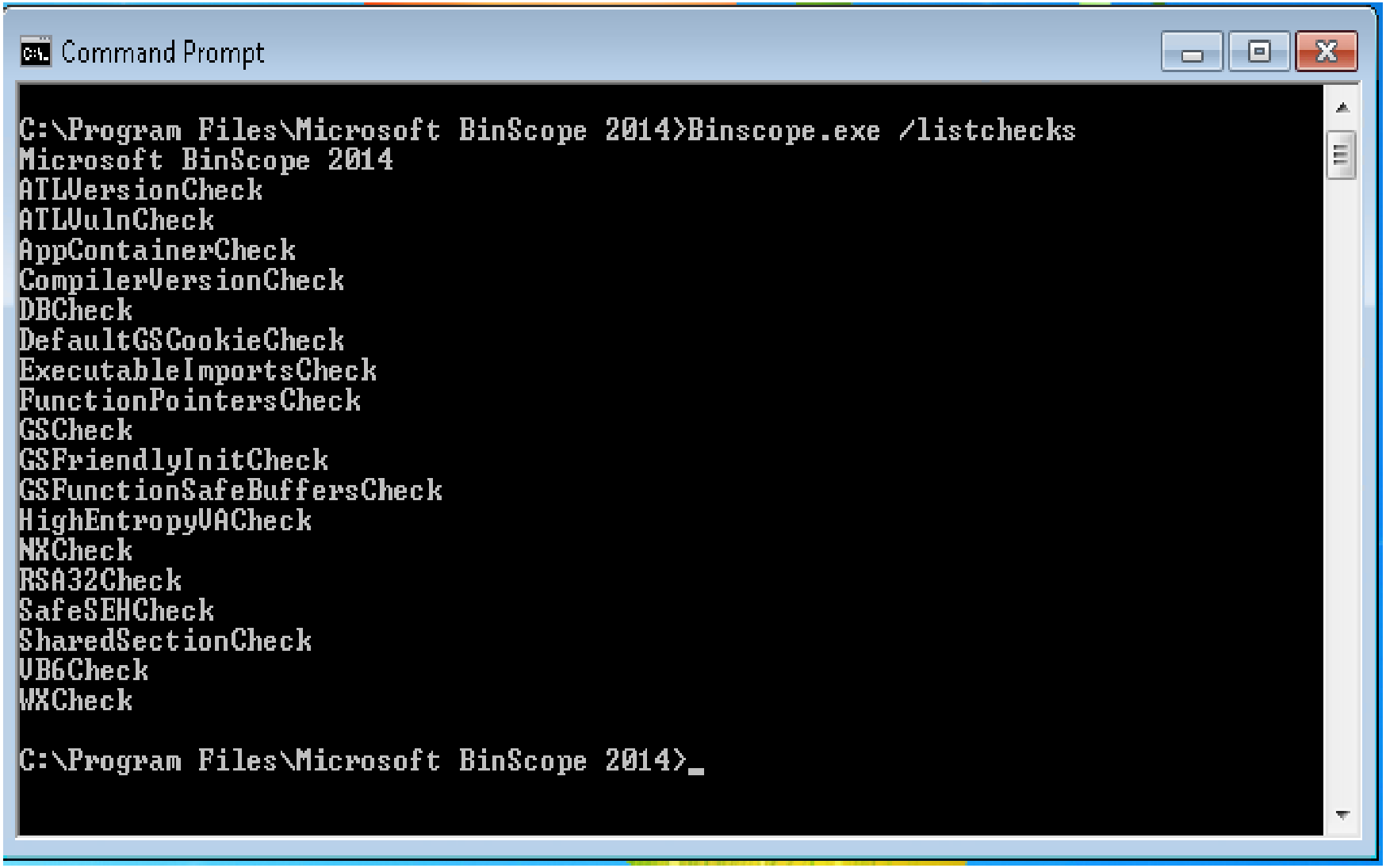
2. TOOL: BINSCOPE

BINSCOPE CHECK	SDL
NXCheck	YES
RSA32Check	YES
SafeSEHCheck	YES
SharedSectionCheck	YES
VB6Check	YES
WXCheck	YES

2. TOOL: BINSCOPE



2. TOOL: BINSCOPE



VULNEX

2. TOOL: BINSCOPE



VULNEX

2. TOOL: BINSKIM

- Version: 1.3.4
- Binary static analysis tool that provides security and correctness results for Windows portable executables
- Download: https://github.com/Microsoft/binskim

2. TOOL: BINSKIM

RULES
Crypto Errors
Security mitigations enabled
Vulnerable libraries
Etc.

2. TOOL: BINSKIM

- Compilation process:

1. Clone / Download code
2. Load src/BinSkim.sln in Visual Studio 2015
3. Set to release mode
4. Build

2. TOOL: BINSKIM



VULNEX

3. IT Security

VULNEX

3. AVAILABLE IT SECURITY TOOLS

1. SECURITY ESSENTIALS / WINDOWS DEFENDER
2. MBSA
3. Microsoft Security Assessment Tool
4. Microsoft Security Compliance Manager
5. WACA
6. Attack Surface Analyzer
7. Portqry
8. EMET
9. Message Analyzer

VULNEX

3. TOOL: SECURITY ESSENTIALS / WINDOWS DEFENDER

- Version: Windows 10
- Identifies and remove malware
- Security Essentials or Windows Defender:
  - Windows 7, Vista and XP: Windows Defender only removes spyware. You must install Security Essentials
  - Windows 8 or later: Windows Defender by default in OS, removes malware
- Download: http://windows.microsoft.com/es-es/windows/security-essentials-download

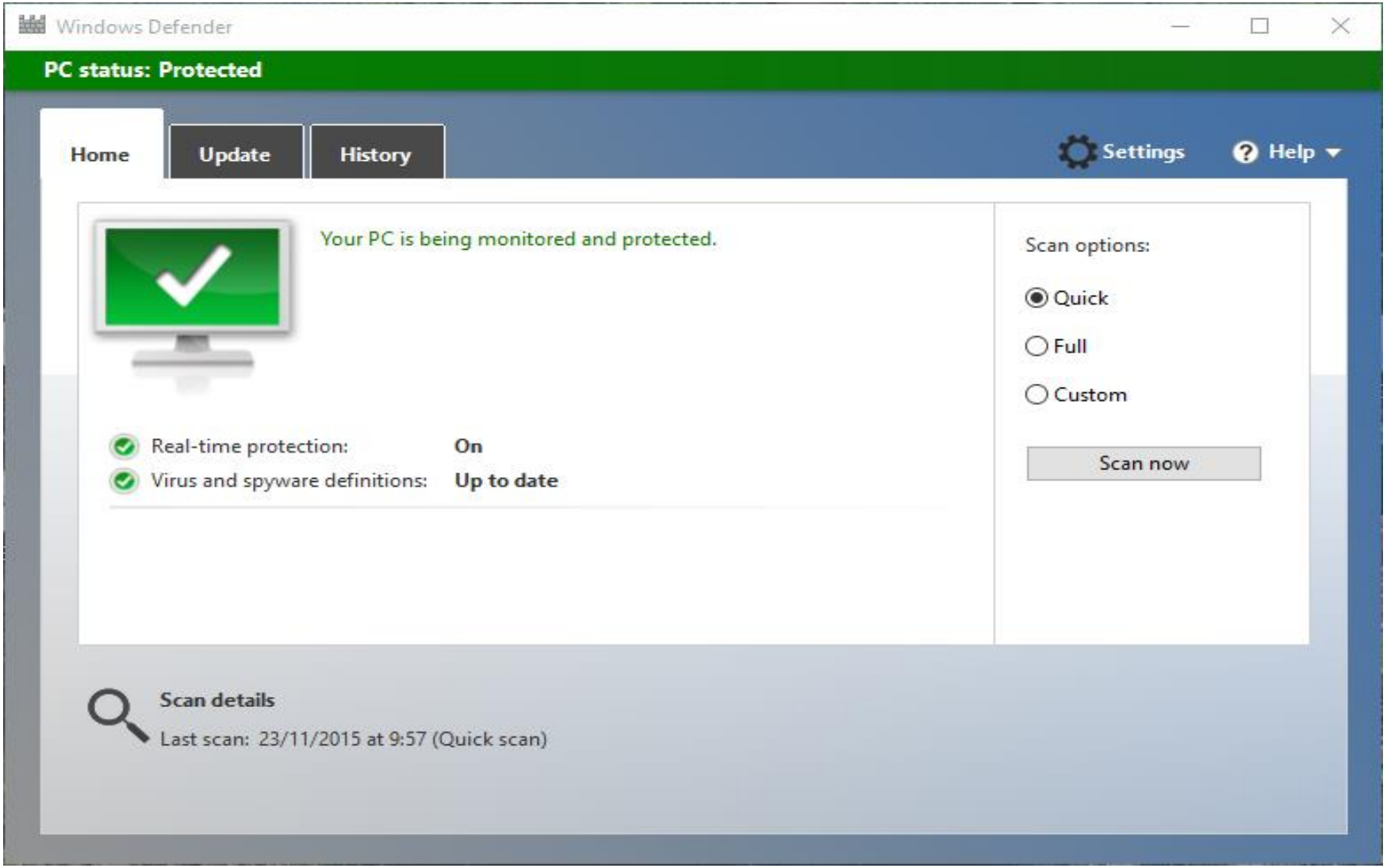
VULNEX

3. TOOL: SECURITY ESSENTIALS / WINDOWS DEFENDER

	Microsoft Security Essentials Windows 7	**Windows Defender** Windows 8,Windows RT, Windows 8.1, Windows RT 8.1, Windows 10
Real-time protection against spyware, viruses, rootkits, and other malicious software	X	X
Online system scanning and cleaning	X	X
Dynamic signature service	X	X
Offline system scanning and cleaning	X	X
Enhanced protection against rootkits and bootkits		X

VULNEX

3. TOOL: SECURITY ESSENTIALS / WINDOWS DEFENDER



VULNEX

3. TOOL: SECURITY ESSENTIALS / WINDOWS DEFENDER



VULNEX

3. TOOL: MBSA

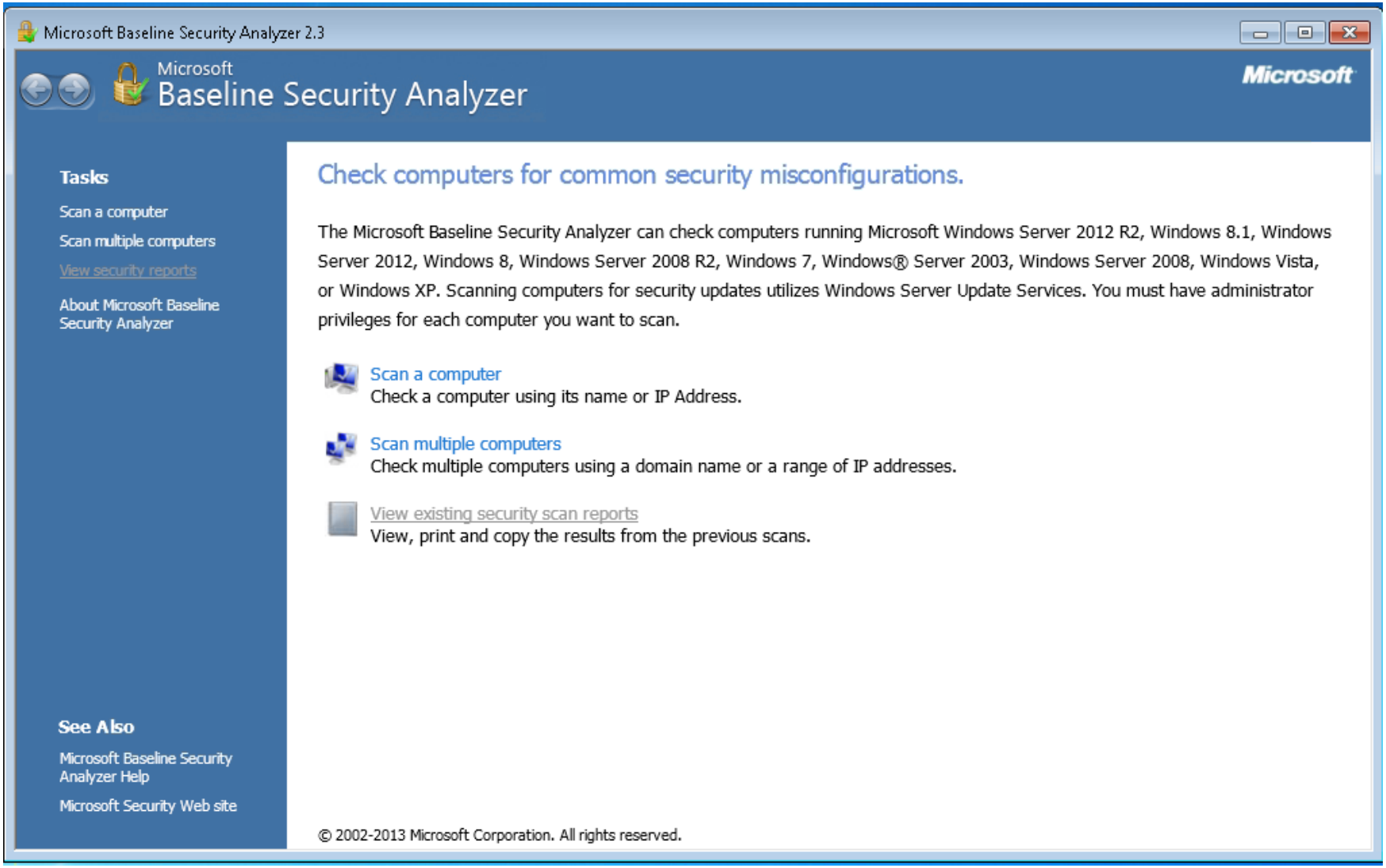
- Version: 2.3
- Microsoft Baseline Security Analyzer (MBSA)
- Security scanner for Windows
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=7558

VULNEX

3. TOOL: MBSA

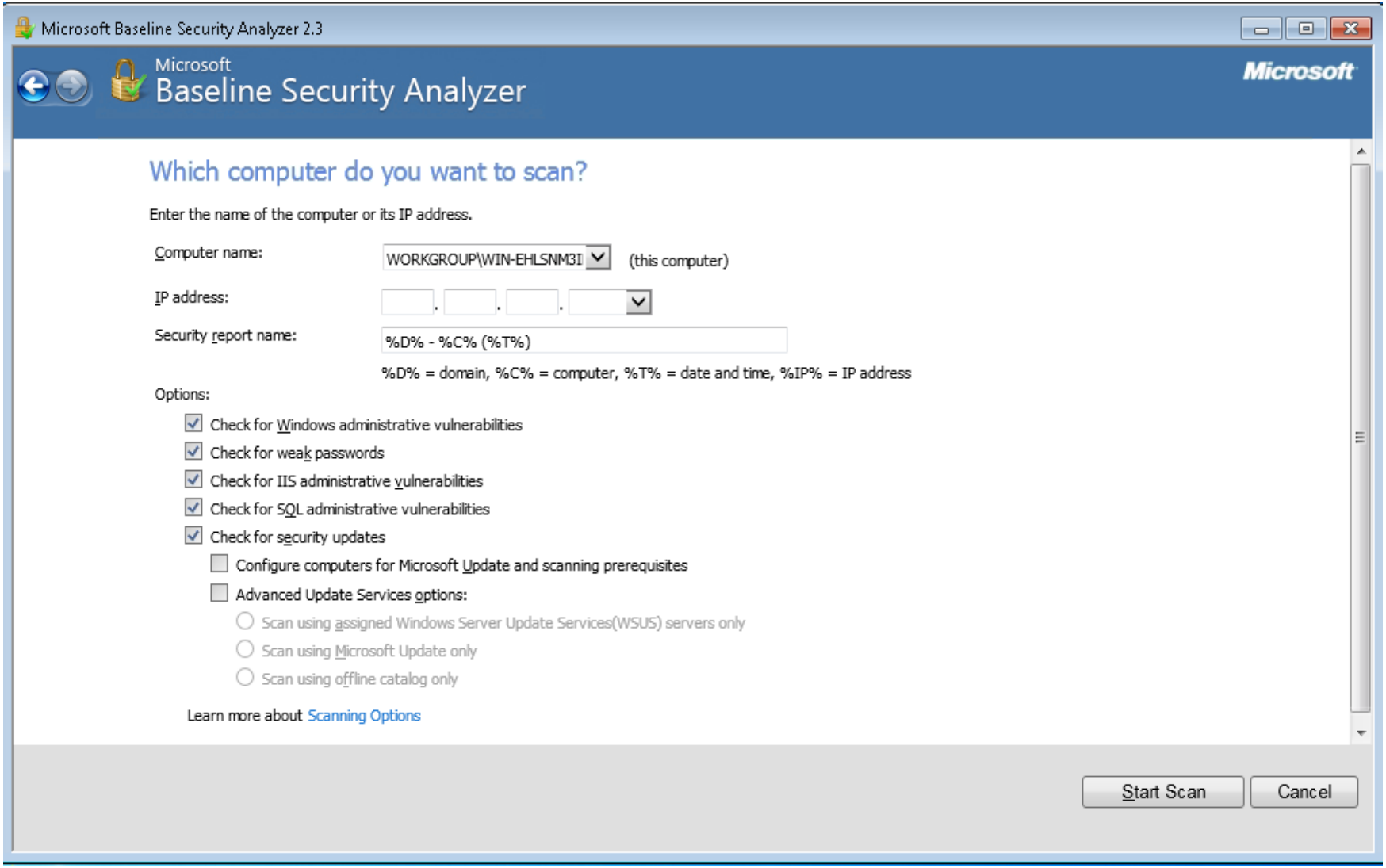
- Scans for:
  - Windows administration vulnerabilities
  - Weak passwords
  - IIS administration vulnerabilities
  - SQL administrative vulnerabilities
- Can configure Windows Update on scanned systems

3. TOOL: MBSA



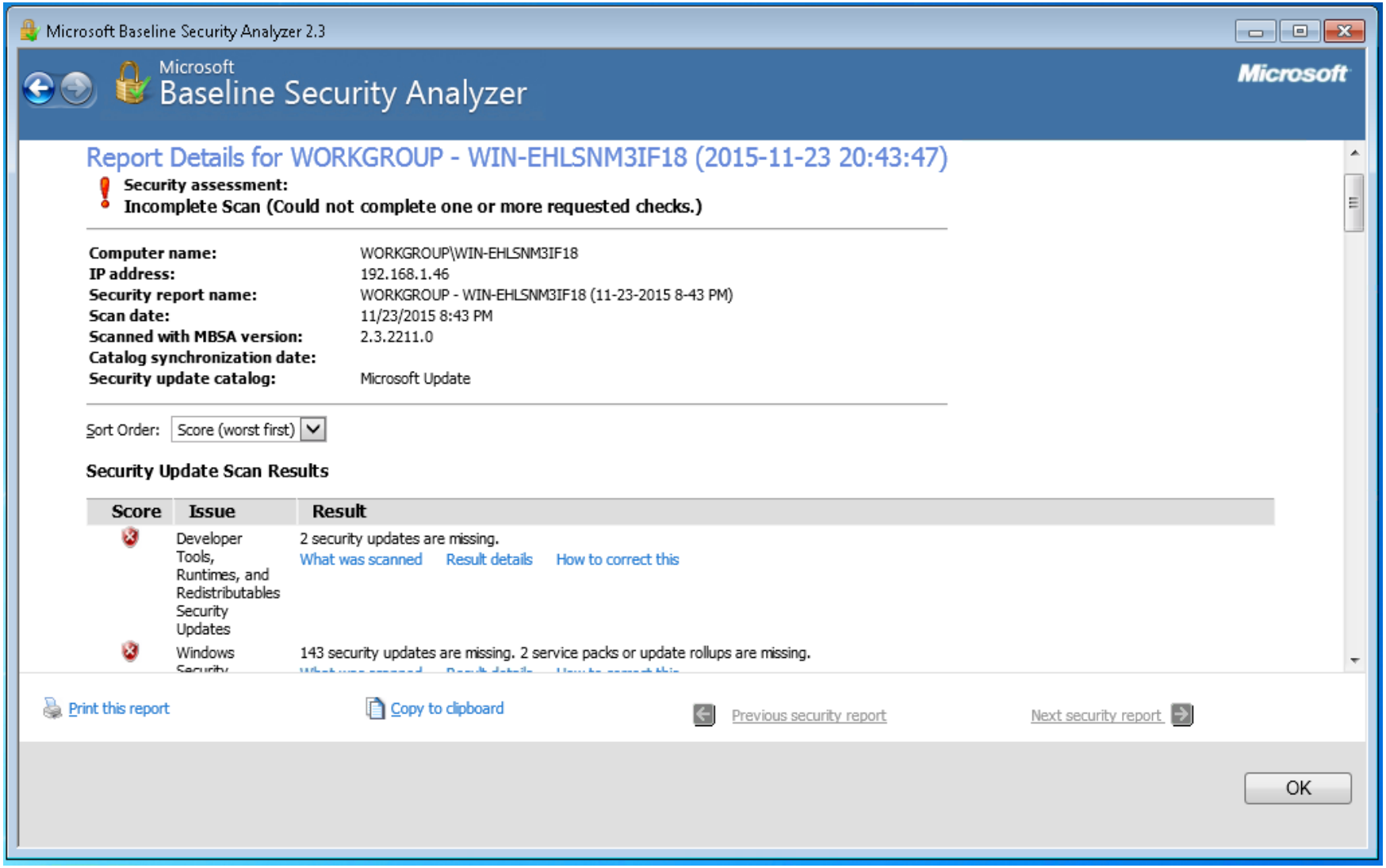
VULNEX

3. TOOL: MBSA



VULNEX

3. TOOL: MBSA



VULNEX

3. TOOL: MBSA



VULNEX

3. TOOL: MICROSOFT SECURITY ASSESSMENT TOOL

- Version: 4.0
- Risk-assessment application designed to provide information and recommendations about best practices for security within an information technology (IT) infrastructure
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=12273

VULNEX

3. TOOL: MICROSOFT SECURITY ASSESSMENT TOOL



VULNEX

3. TOOL: MICROSOFT SECURITY COMPLIANCE MANAGER

- Version: 3.0
- Provides centralized security baseline management features, a baseline portfolio, customization capabilities, and security baseline export flexibility to accelerate your organization's ability to efficiently manage the security and compliance process for the most widely used Microsoft technologies
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=16776

VULNEX

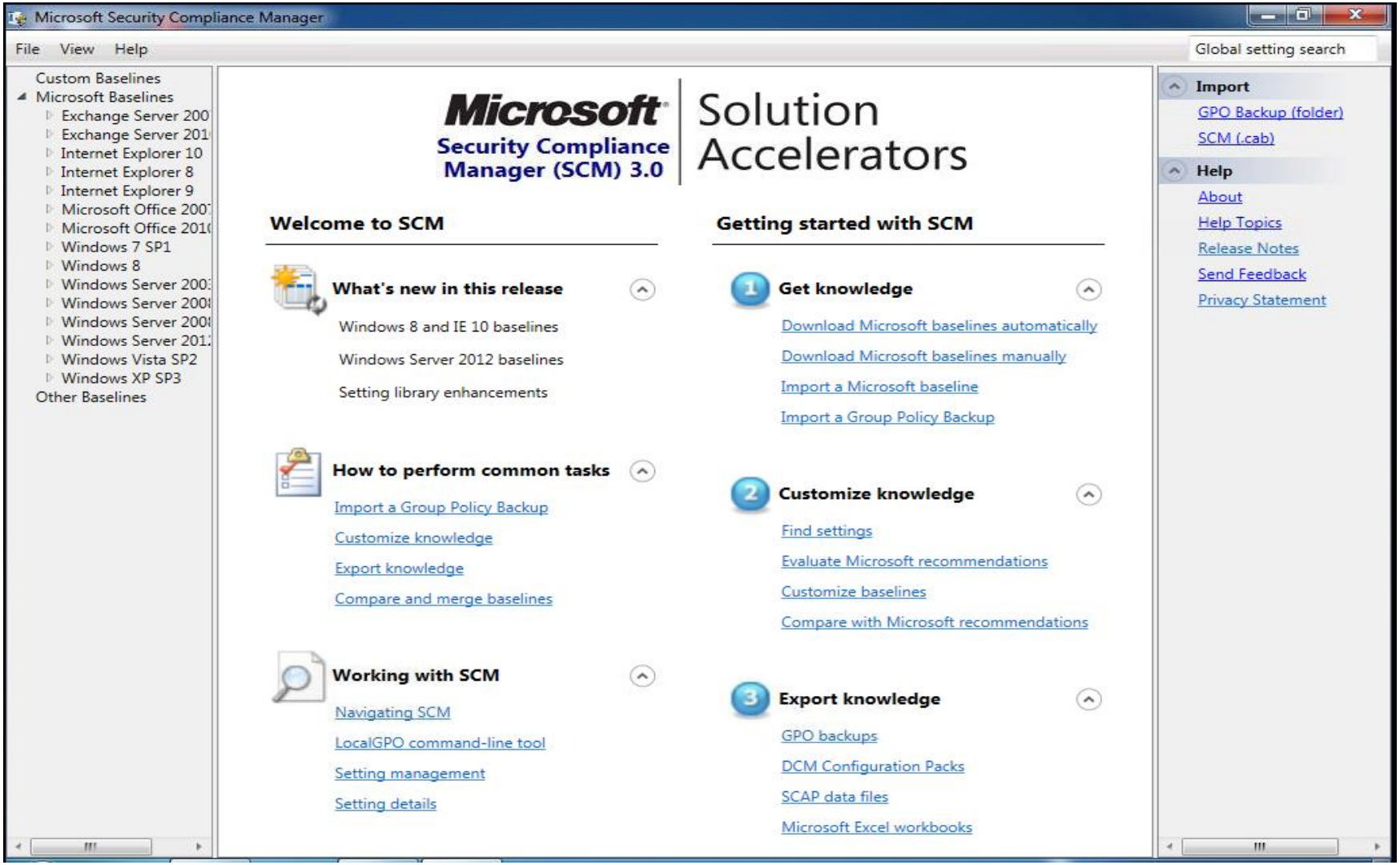
3. TOOL: MICROSOFT SECURITY COMPLIANCE MANAGER

- Note: SCM Version 3.0 do not install on Windows 10 due to incompatible SQL Server 2008 Express

  However if you install SQL Server 2008 R2 Express Edition standalone, you then can install SCM in Windows 10

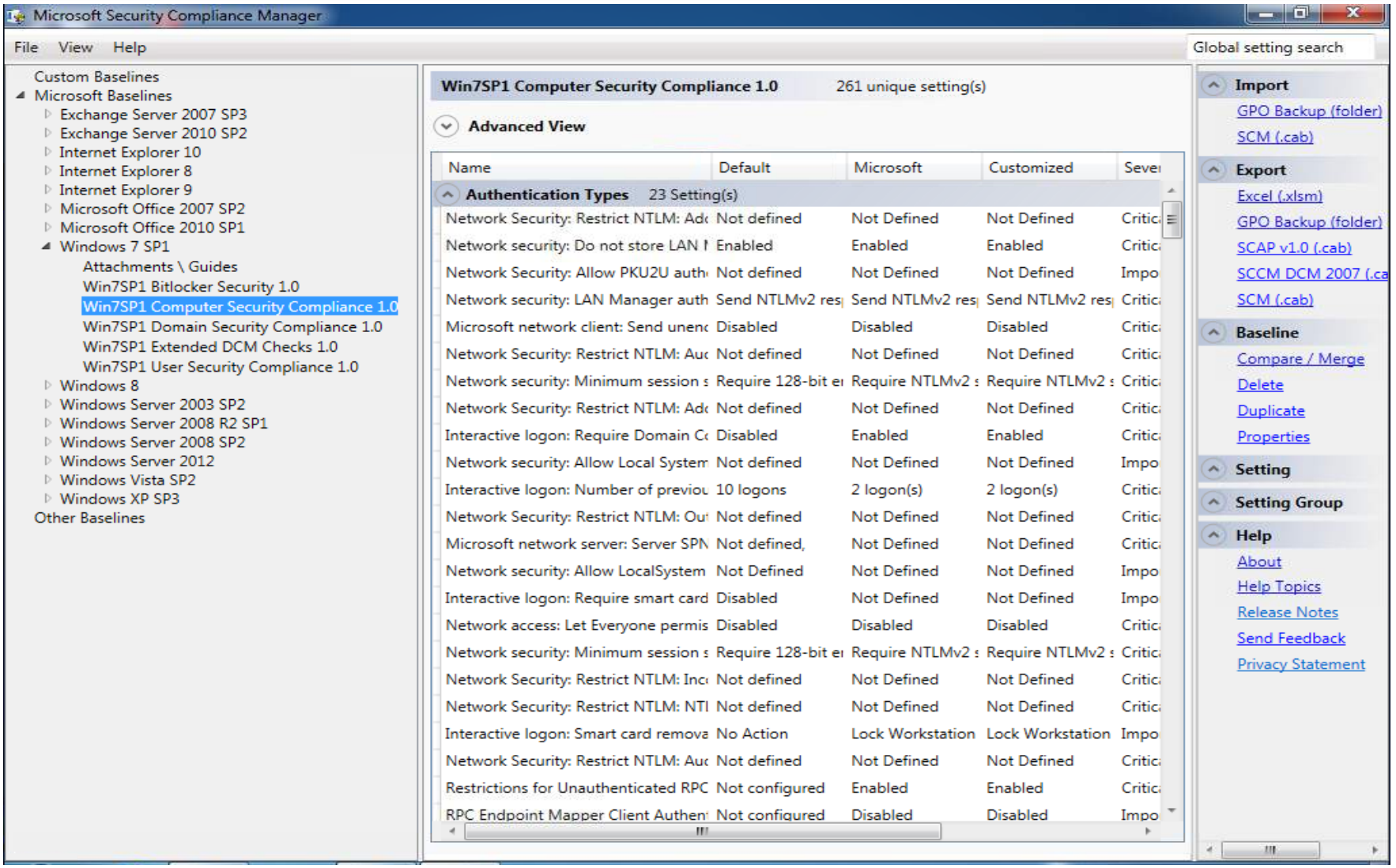
  https://www.microsoft.com/en-US/download/details.aspx?id=30438

3. TOOL: MICROSOFT SECURITY COMPLIANCE MANAGER



VULNEX

3. TOOL: MICROSOFT SECURITY COMPLIANCE MANAGER



VULNEX

3. TOOL: MICROSOFT SECURITY COMPLIANCE MANAGER



VULNEX

3. TOOL: WACA

- Version: 2.0
- Microsoft Web Application Configuration Analyzer
- Download: http://www.microsoft.com/en-us/download/details.aspx?id=573

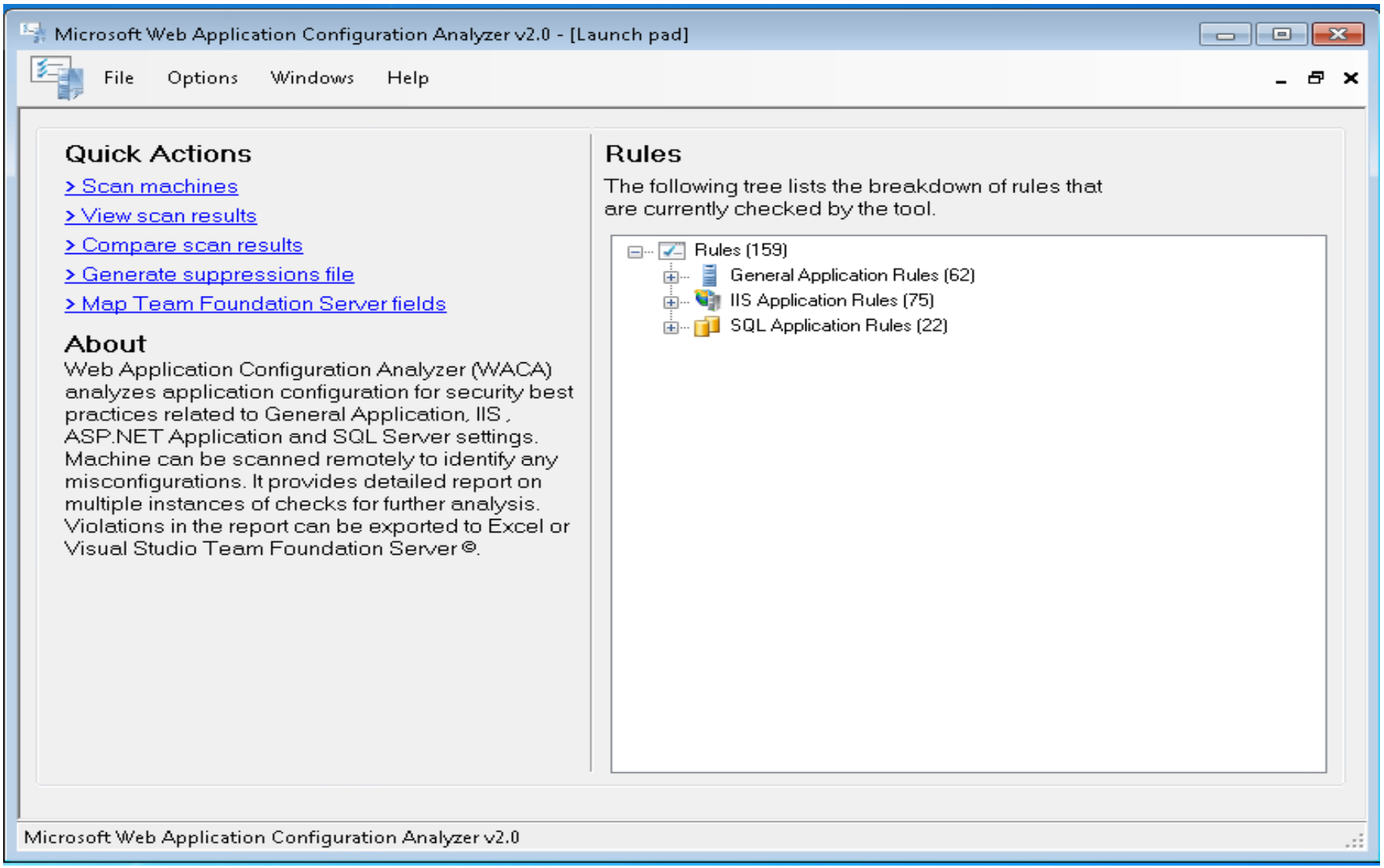
VULNEX

3. TOOL: WACA

WACA RULES
General Application Rules (62)
IIS Application Rules (75)
SQL Application Rules (22)

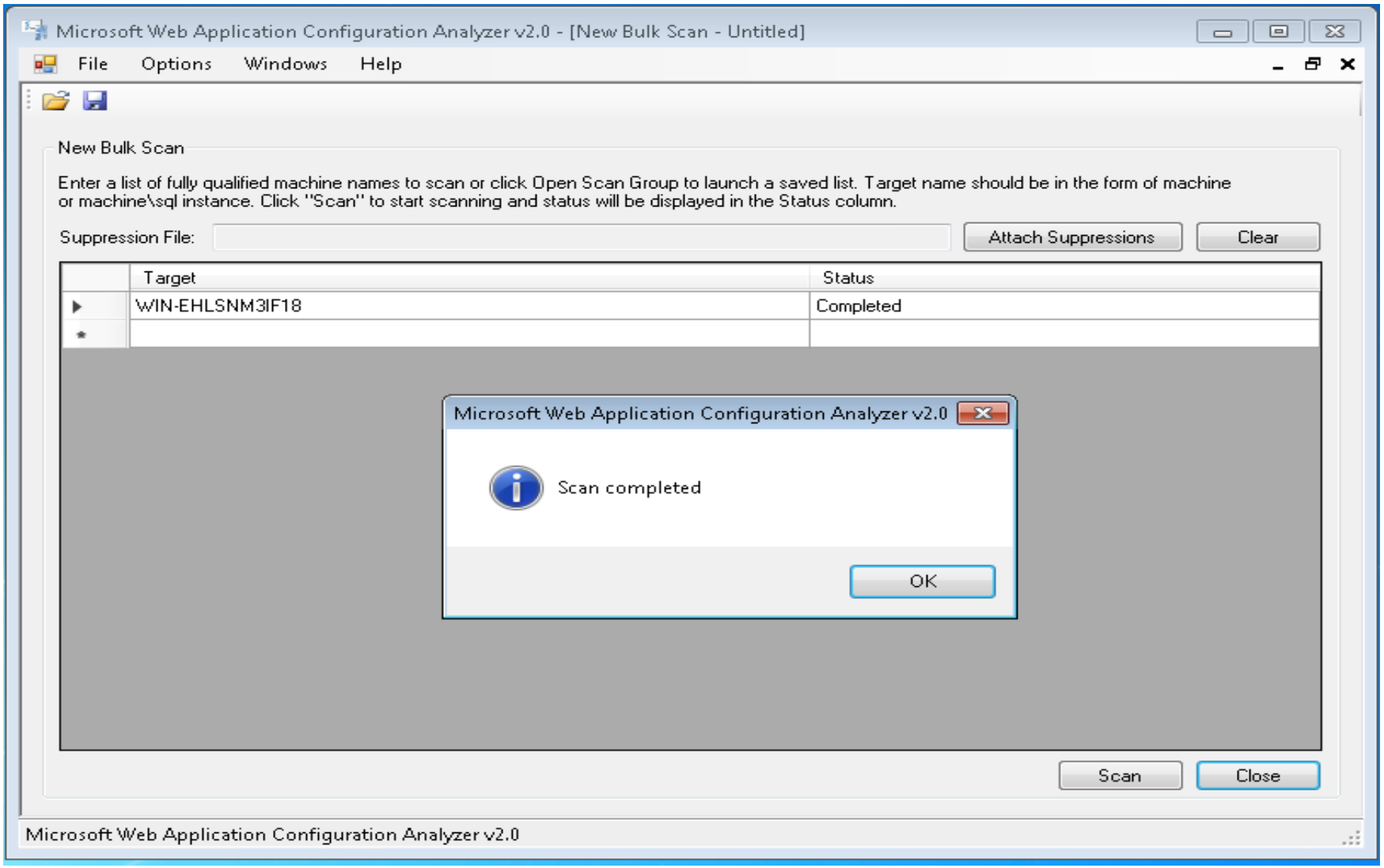
VULNEX

3. TOOL: WACA



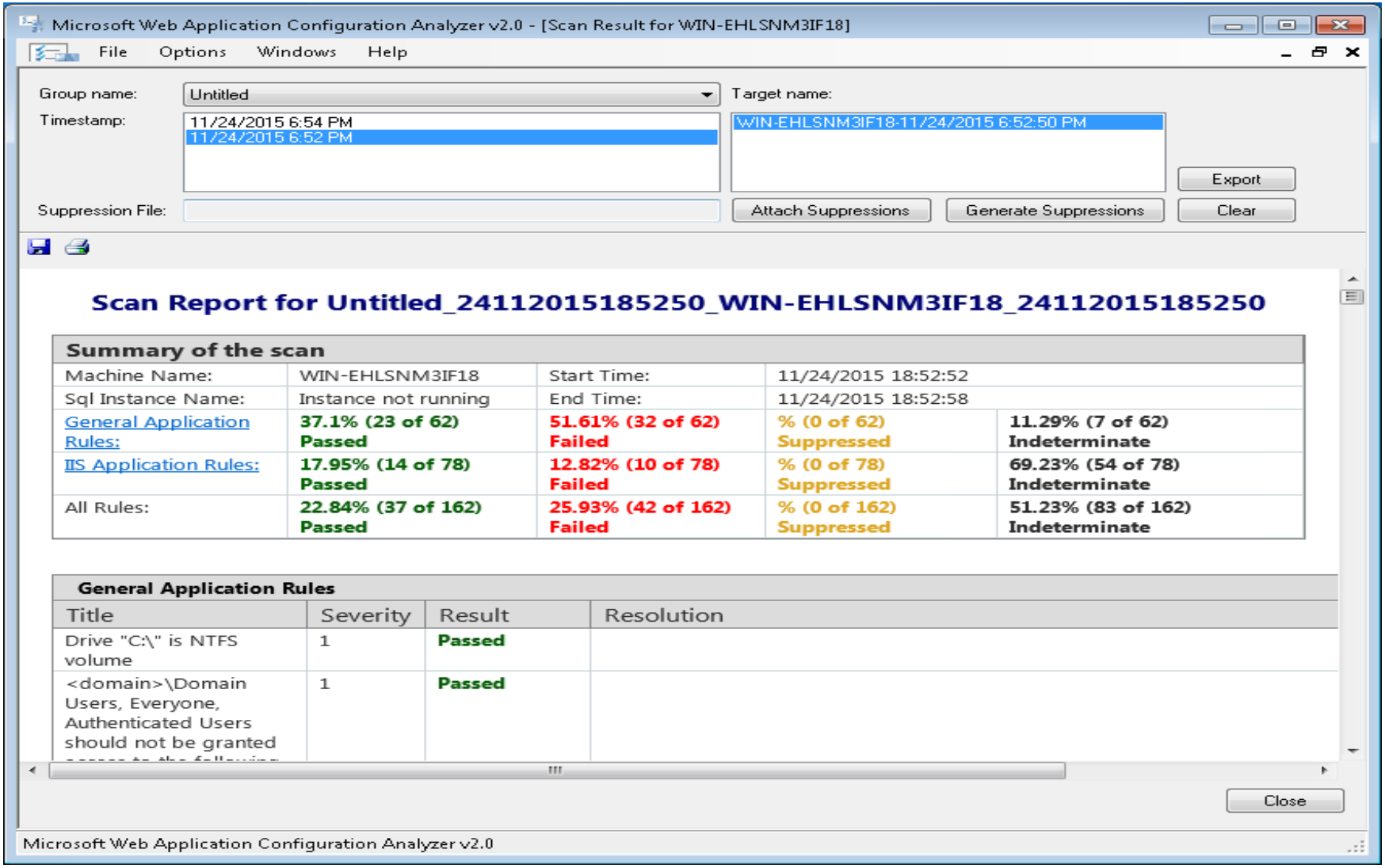
VULNEX

3. TOOL: WACA



VULNEX

3. TOOL: WACA



VULNEX

3. TOOL: WACA



VULNEX

3. TOOL: ATTACK SURFACE ANALYZER

- Version: 1.0
- Identifies changes to a Windows system when installing an application
- Ideally run on a system equal to production
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=24487

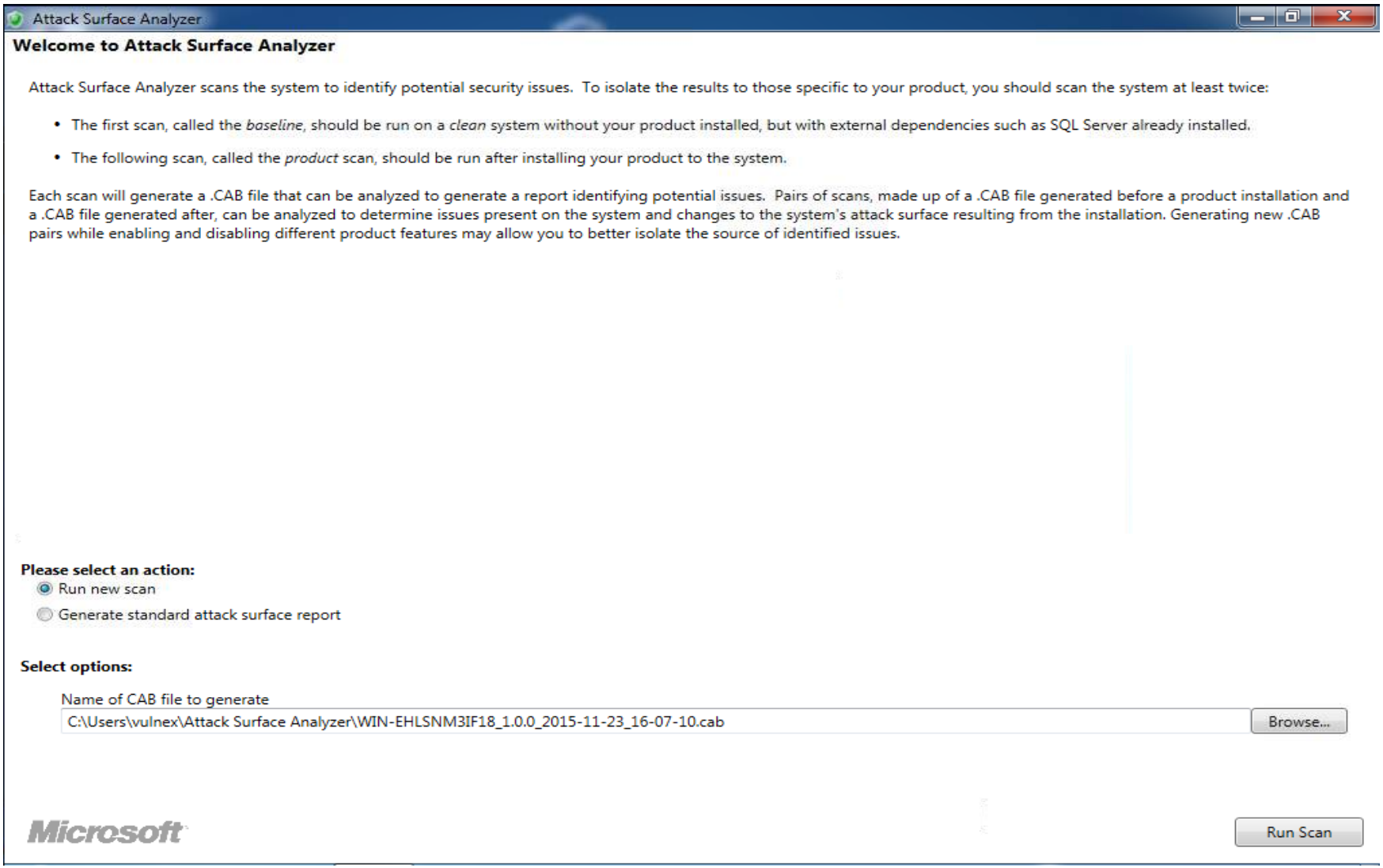
VULNEX

3. TOOL: ATTACK SURFACE ANALYZER

SCANS FOR
Registry
File Systems
Registered Filetypes
Ports
Process
Etc.

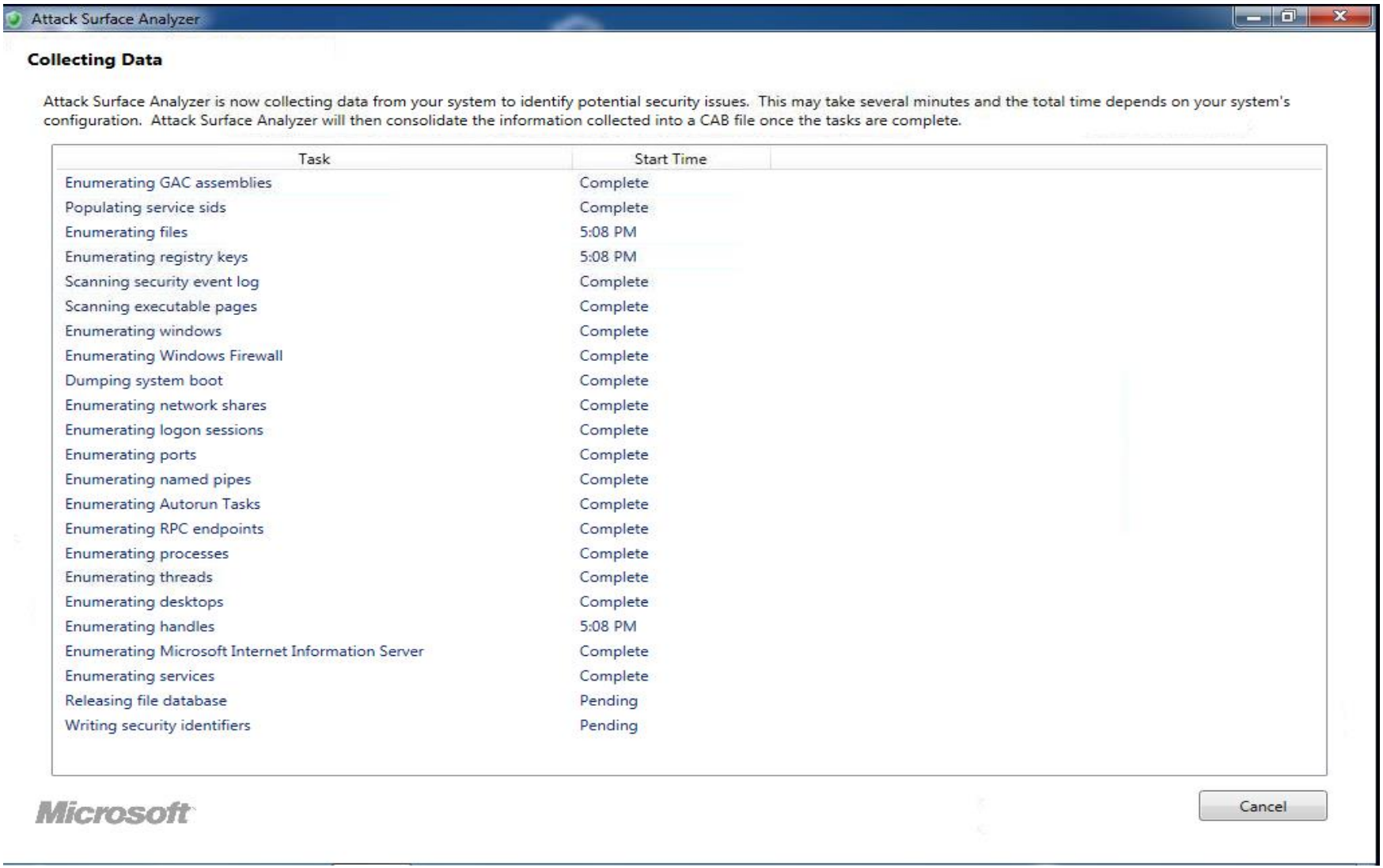
VULNEX

3. TOOL: ATTACK SURFACE ANALYZER



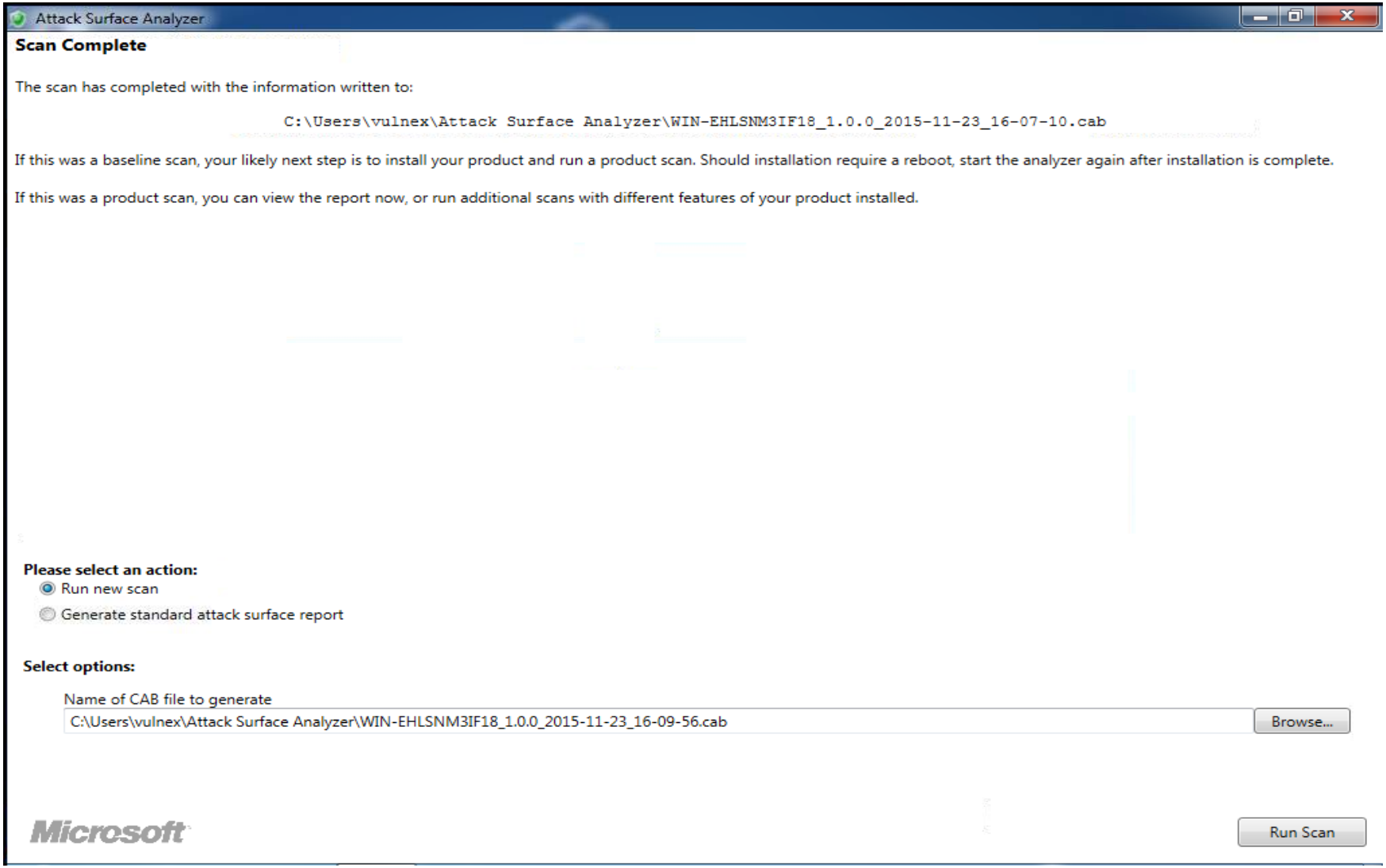
VULNEX

3. TOOL: ATTACK SURFACE ANALYZER



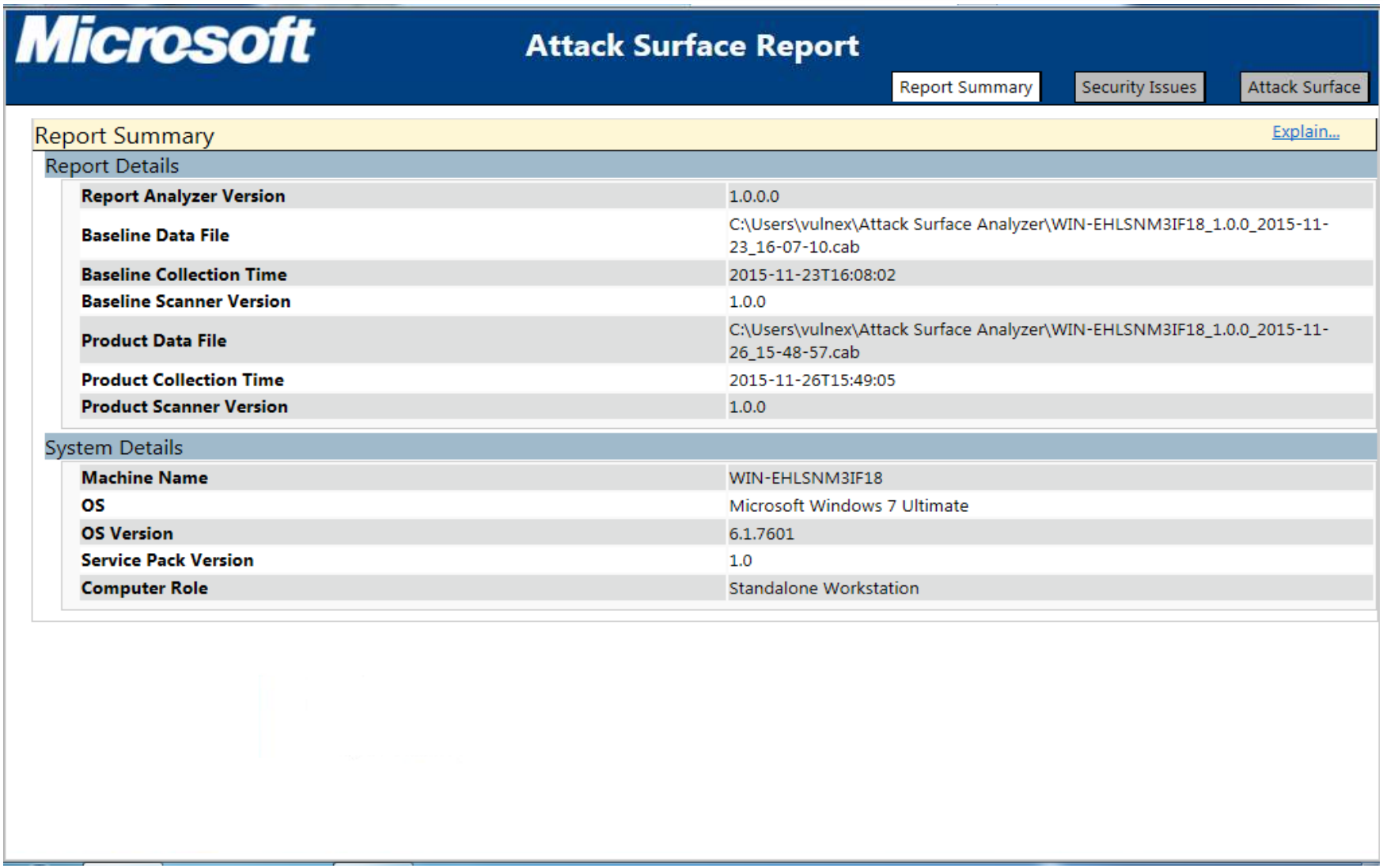
VULNEX

3. TOOL: ATTACK SURFACE ANALYZER



VULNEX

3. TOOL: ATTACK SURFACE ANALYZER



Microsoft

Attack Surface Report

Report Summary | Security Issues | Attack Surface

Report Summary — Explain...

Report Details

Report Analyzer Version	1.0.0.0
Baseline Data File	C:\Users\vulnex\Attack Surface Analyzer\WIN-EHLSNM3IF18_1.0.0_2015-11-23_16-07-10.cab
Baseline Collection Time	2015-11-23T16:08:02
Baseline Scanner Version	1.0.0
Product Data File	C:\Users\vulnex\Attack Surface Analyzer\WIN-EHLSNM3IF18_1.0.0_2015-11-26_15-48-57.cab
Product Collection Time	2015-11-26T15:49:05
Product Scanner Version	1.0.0

System Details

Machine Name	WIN-EHLSNM3IF18
OS	Microsoft Windows 7 Ultimate
OS Version	6.1.7601
Service Pack Version	1.0
Computer Role	Standalone Workstation

VULNEX

3. TOOL: ATTACK SURFACE ANALYZER

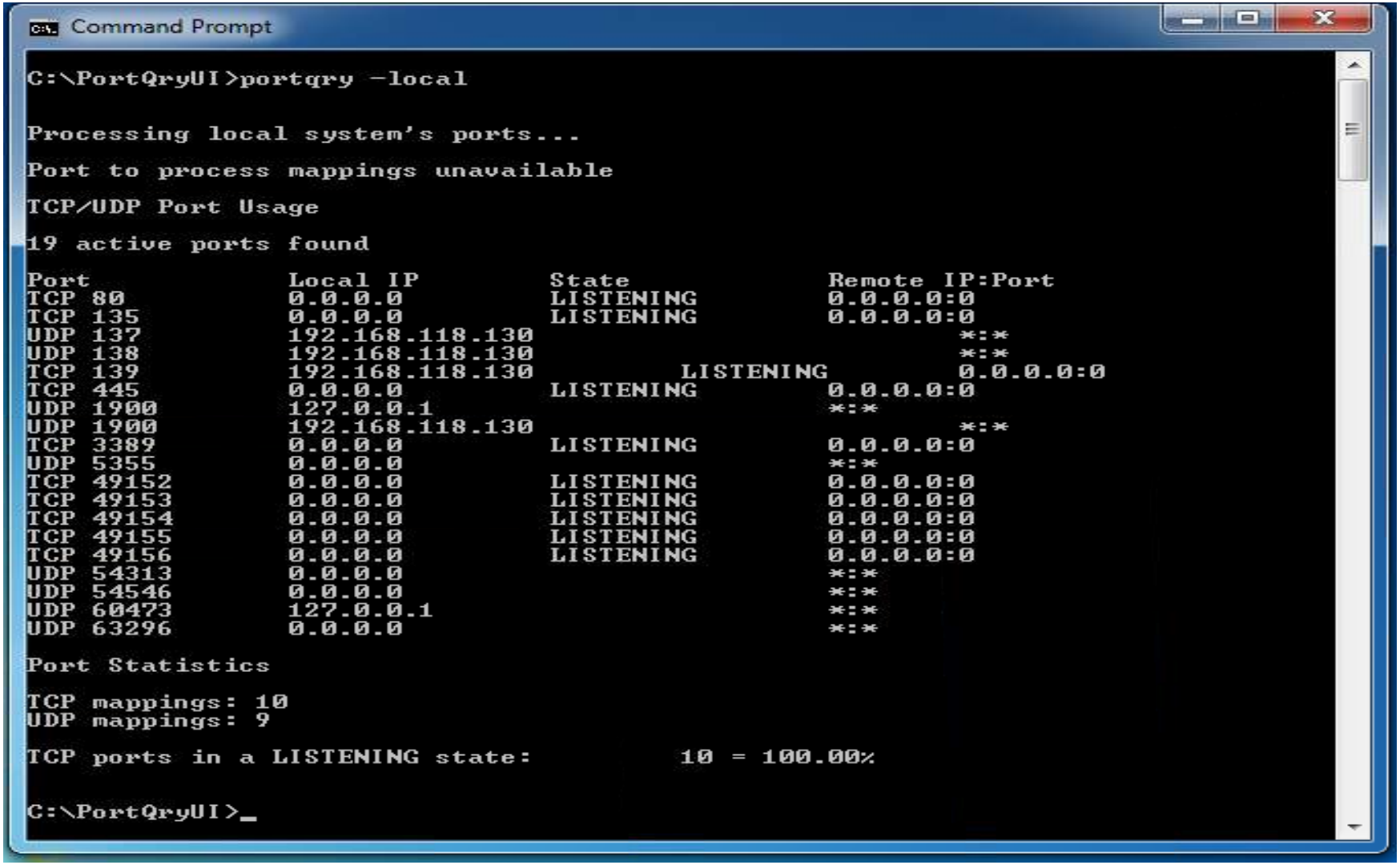


VULNEX

3. TOOL: PORTQRY

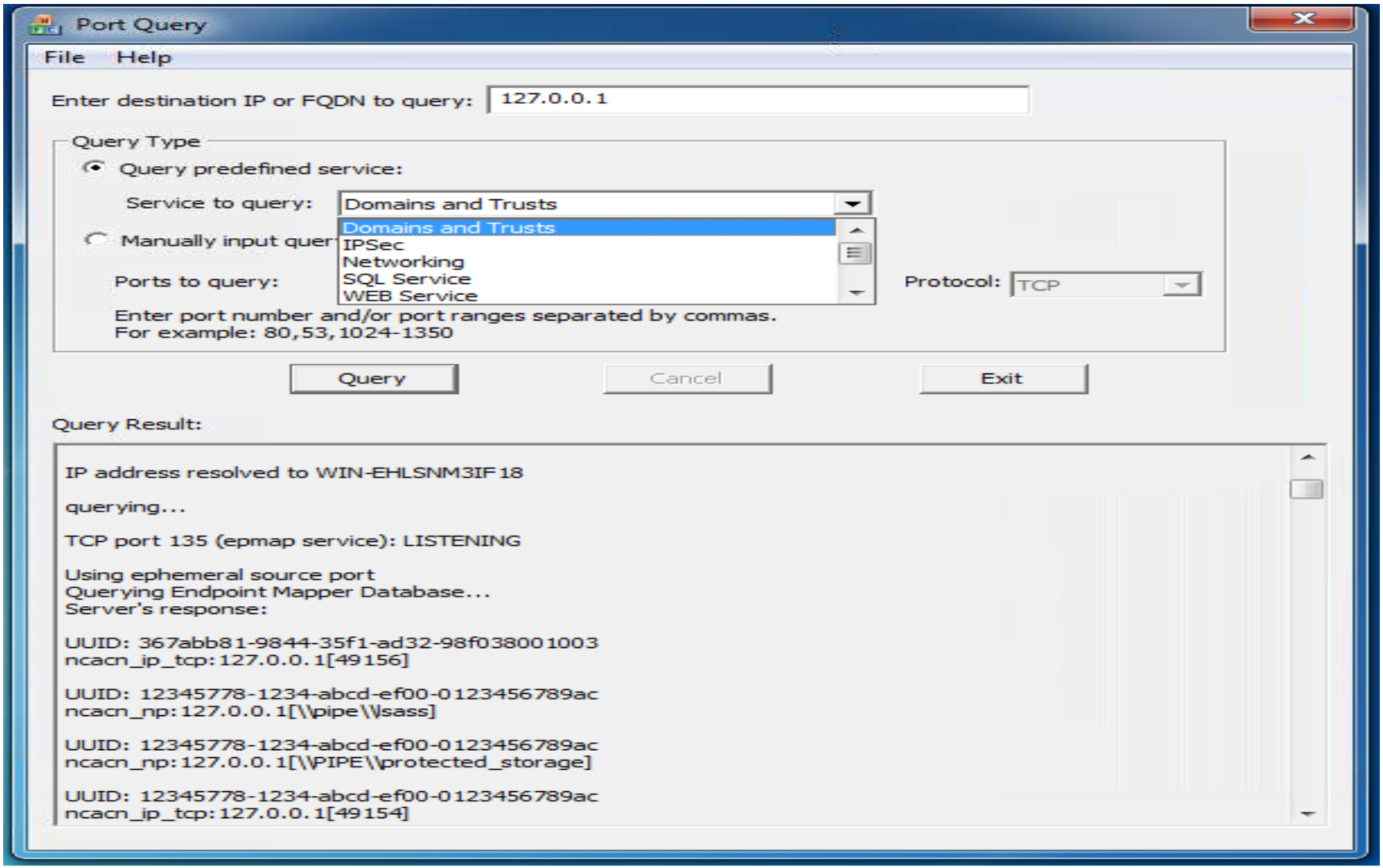
- Version: 2.0
- Port scanner
- GUI and command line
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=24009

3. TOOL: PORTQRY



VULNEX

3. TOOL: PORTQRY



VULNEX

3. TOOL: PORTQRY



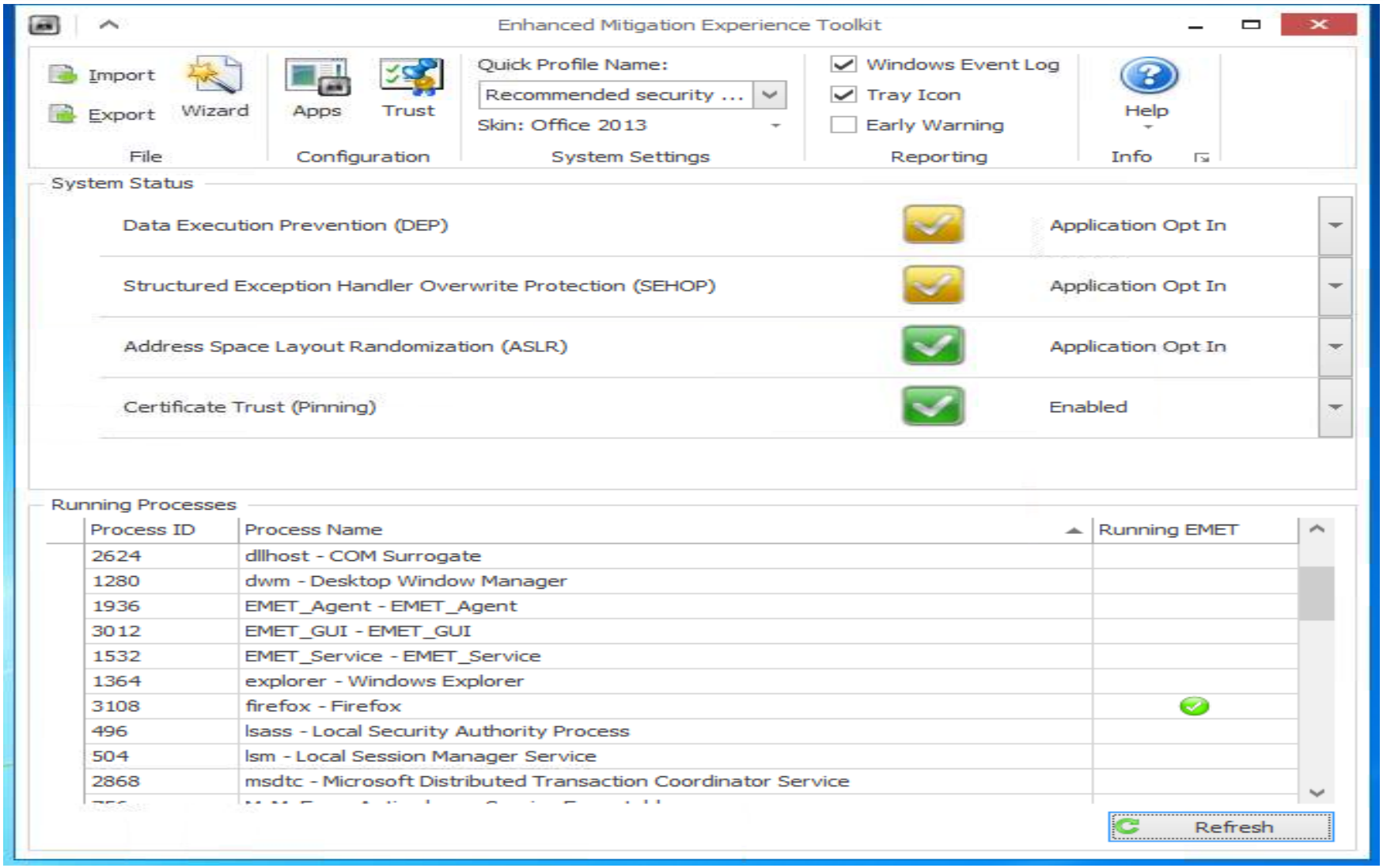
VULNEX

3. TOOL: EMET

- Version: 5.5
- Enhanced Mitigation Experience Toolkit (EMET)
- Toolkit for deploying and configuring security mitigation technologies
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=46366

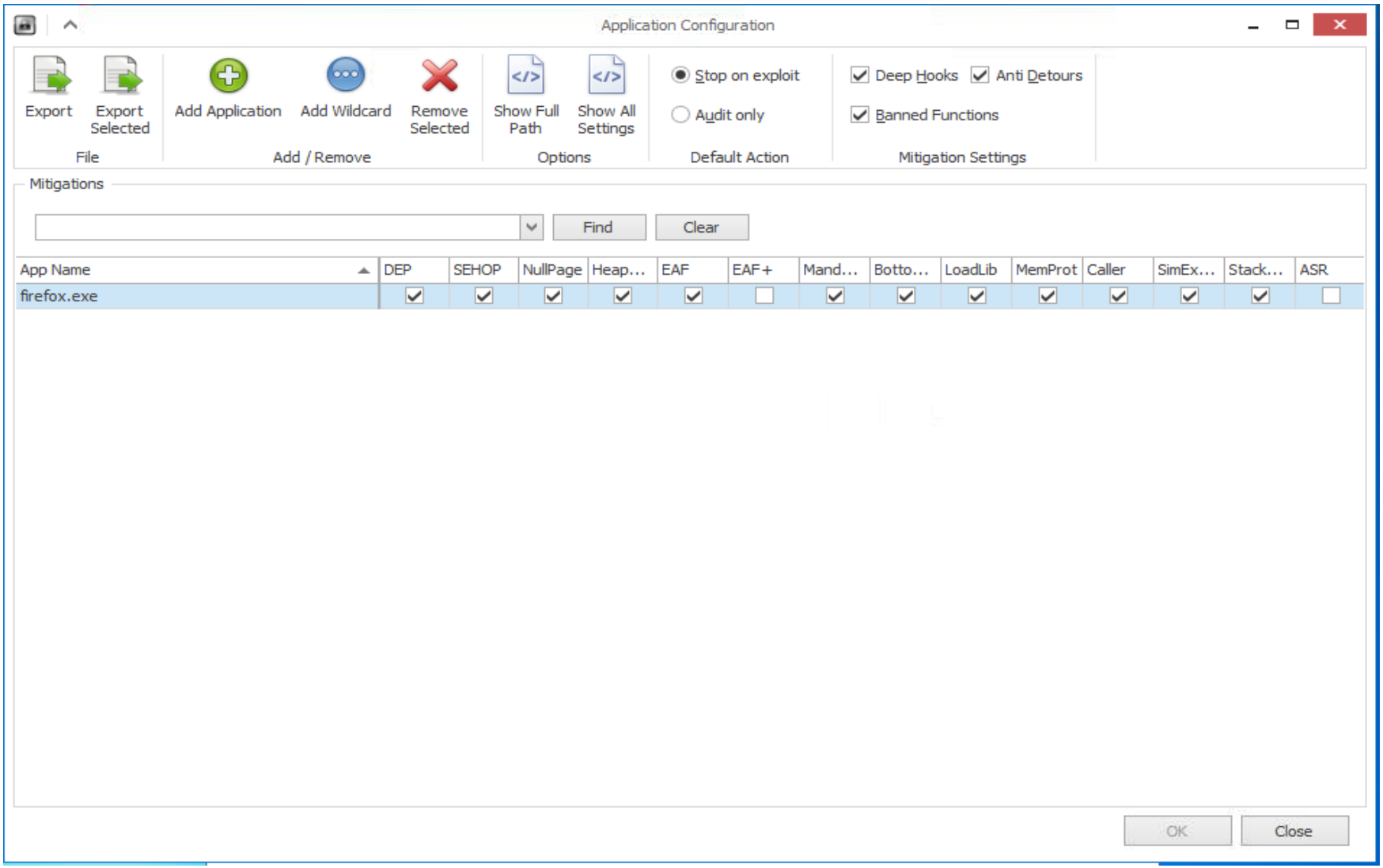
VULNEX

3. TOOL: EMET



VULNEX

3. TOOL: EMET



VULNEX

3. TOOL: EMET



VULNEX

3. TOOL: MESSAGE ANALYZER

- Version: 1.4
- Enables to capture, display, and analyze protocol messaging traffic; and to trace and assess system events and other messages from Windows components
- Download: https://www.microsoft.com/en-us/download/details.aspx?id=44226

3. TOOL: MESSAGE ANALYZER



VULNEX

4. Conclusions

VULNEX

4. SECURITY ARSENAL

- A vast arsenal of free security tools released by Microsoft (Thanks):

  1. Development
  2. TI

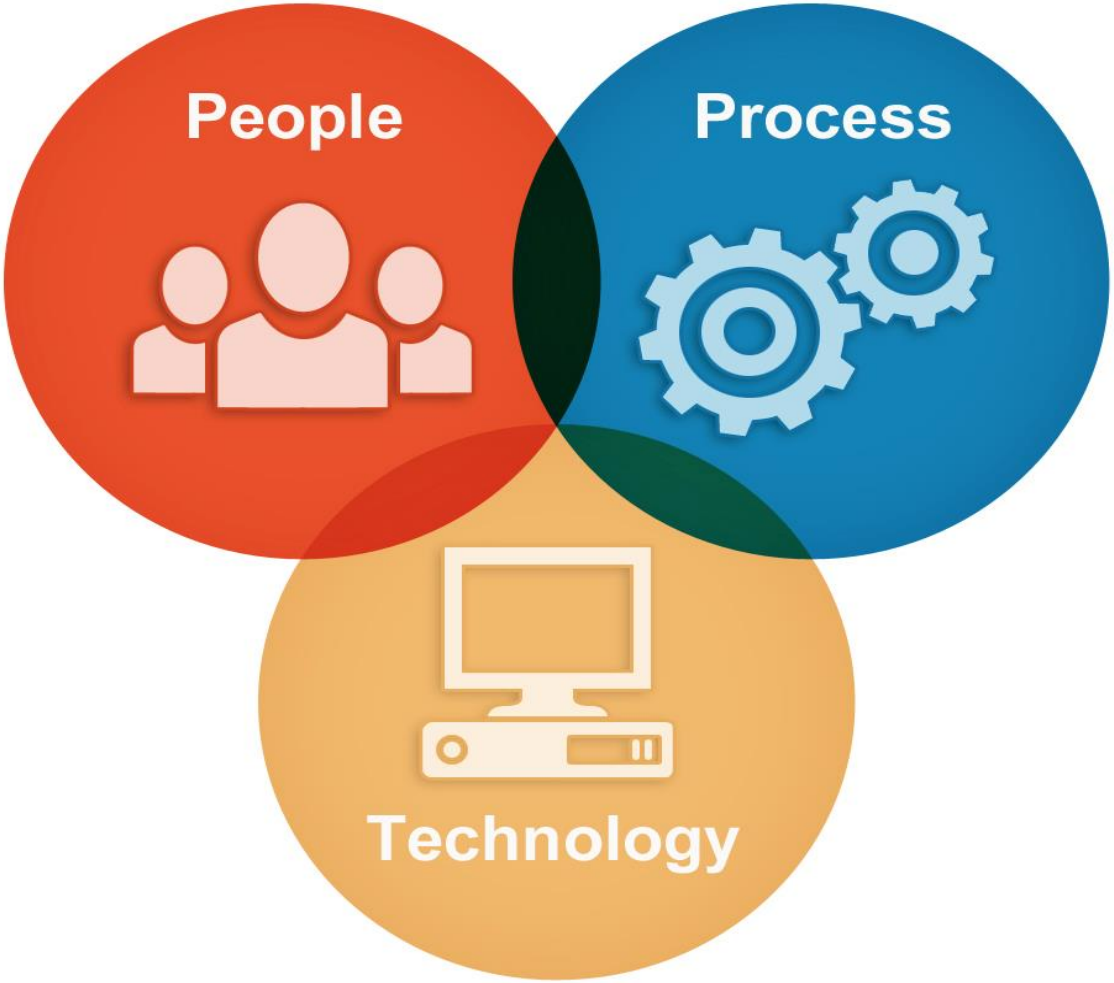
- There is even more tools available!

4. NO EXCUSES



VULNEX

4. ONLY TECHNOLOGY IS NOT ENOUGH



VULNEX

4. FREE TRAINING

- Microsoft SDL Process Training
  https://www.microsoft.com/en-us/sdl/process/training.aspx

- SAFECode Training
  https://training.safecode.org/

VULNEX

5. Q&A

- Thanks!
- Beer appreciated!!!
-  @simonroses
-  @vulnexsl
- www.vulnex.com
- www.simonroses.com

VULNEX