

.NET from the Hacker's Perspective

Drew Miller

Drew.miller@securityage.com

.NET from the Hacker's Perspective

What Hackers Dislike

Risk

What Hackers Like

Summary

This presentation entitled “.NET from a hacker's perspective” focuses on specific .NET technologies on a high-level to show how hackers are approaching this new technology.

What Hackers Dislike

- **.NET Buffer Overflows**
- Role Security
- CAS Code Access Security
- Cryptography
- Summary

Many potential vulnerabilities have been addressed by .NET. We focus on a few of the technological advances that .NET has made and how it helps improve security.

.NET Buffer Overflows

- Managed Code
- Legacy Code
- The Developer Mind Set

Buffer overflows are a major concern for all legacy applications for both desktop and network use. We will dive into how .NET keeps buffer overflows from occurring and where it doesn't.

.NET Buffer Overflows: Managed Code

- Self-resizing variables
- .NET Framework keeps fixed sized variables from being copied to by variable sized variables

The .NET Framework will not allow fixed sized variables to be written to by variable length variables. The compiler will not let you compile a program that attempts to do this in code. Strings which are commonly used everywhere, especially in input fields now have the keen ability to always resize themselves.

.NET Buffer Overflows: Legacy Code

- It is still very common to use previously coded modules and routines
- Why reinvent the wheel?
 - Security?

However, no one wants to rewrite the last ten (10) years of production code, so many modules and routines from previous work is being included through the framework's interoperability. This interoperability though very flexible and great for development, leaves new applications still vulnerable to one of the oldest attacks in the book.

.NET Buffer Overflows: The Developer's Mind Set

- No buffer overflows in .NET? I no longer need to bounds check my variable length variables.
- Less could mean more

A common mistake for new .NET development is the belief that bounds checking is no longer required. The moment you included that legacy COM+ module... it's buffer overflows all over again.

What Hackers Dislike

- Buffer Overflows
- **Role Security**
- CAS Code Access Security
- Cryptography
- Summary

Many potential vulnerabilities have been addressed by .NET. We focus on a few of the technological advances that .NET has made and how it helps improve security.

Role Security

- Don't call me... I'll call you
- Framework for defining class and function level call security

Role security in .NET allows to keep an a current process that has an associated identity from calling or executing code paths that require a privilege level which the current identity does not have.

What Hackers Dislike

- Buffer Overflows
- Role Security
- CAS Code Access Security
- Cryptography
- Summary

Many potential vulnerabilities have been addressed by .NET. We focus on a few of the technological advances that .NET has made and how it helps improve security.

CAS Code Access Security

- Mobile Code
- Default user permission settings for the Internet Zone makes hard case for ignoring use in public market
- Signing Assemblies (GAC)
- Key Management (Source Safe)

Code access security gives the developer many features including; a way to stamp the code to avoid/detect Trojan attempts, distribute code that has less permissions than ActiveX (“All Or Nothing Controls”), versioning code cryptographically.

What Hackers Dislike

- Buffer Overflows
- Role Security
- CAS Code Access Security
- **Cryptography**
- Summary

Many potential vulnerabilities have been addressed by .NET. We focus on a few of the technological advances that .NET has made and how it helps improve security.

Cryptography

- Encrypt vs. Encode vs. Hashing
- Minimal Coding Requirements
- Fast
- Easy Key Management
- XML

Cryptography used to be much more difficult to code. With .NET the source code examples tell the whole story. Less than ten (10) lines of code a required to create keys, encrypt and decrypt... and with resizing `byte[]` (byte arrays), no more wild memory allocation routines are required. Crypto is for everyone. Start using it now.

What Hackers Dislike

- Buffer Overflows
- Role Security
- CAS Code Access Security
- Cryptography
- **Summary**

Many potential vulnerabilities have been addressed by .NET. We focus on a few of the technological advances that .NET has made and how it helps improve security.

What Hackers Dislike: Summary

- Buffer Overflow Protection
 - Always bounds check
- Role Based Security In Code
 - Validate who is allowed to call functions
- Newer Code Difficult To Trojan
 - Avoid Trojans like “FunLove”
- Everything Encrypted
 - Avoid information leakage

A large amount of exposures and vulnerabilities can instantly be taken care of by including the use of new .NET technologies when developing .NET applications. Including stand-alone applications and web applications.

.NET from the Hacker's Perspective

What Hackers Dislike

Risk

What Hackers Like

Summary

This presentation entitled “.NET from a hacker’s perspective” focuses on specific .NET technologies on a high-level to show how hackers are approaching this new technology.

Risk

- Everyone has a deadline
- Everyone has a performance requirement
- NEW -> Everyone has a security requirement
- Dollar -> Security -> Risk

Risk should always be considered before talking about exposures and vulnerabilities. Someone, somewhere, has to make a decision whether an exposure or vulnerability is TOO much of a risk, and to fix that before software is released to the public, or to a private customer.

.NET from the Hacker's Perspective

What Hackers Dislike
Risk
What Hackers Like
Summary

This presentation entitled “.NET from a hacker’s perspective” focuses on specific .NET technologies on a high-level to show how hackers are approaching this new technology.

What Hackers Like

- **Information Leakage**
 - View state
 - XML
 - SQL errors
 - Web errors
 - Cookies
 - URLs
- Does easy to develop mean easy to exploit?
- Cross Site Scripting
- Reaply/Hijacking
- Injection XML/SQL

What information does your application give a user? How can that be used against you? Small to medium businesses suffer due to time-to-market requirements. Are hackers going after the little people more often due to these circumstances?

Information Leakage: View State

- View State
 - Base64 encoded
 - Dynamic properties of server-side controls
 - Map to exposures and vulnerabilities

The view state can leak information. If a property of a server-side control is dynamic and the view state is enabled then that information is sent to the client and can be viewed by 1) viewing the source of the web page that is generated by .NET and then 2) decoding the view state block which is blatantly marked. It can be encrypted, but if the size is very large, is there more of a performance hit by encrypting it, or turning off the view state altogether.

Information Leakage: XML

- The world of plaintext
- Sniffed traffic can lead to information leakage
- Encrypting XML can be cumbersome and degrades performance
- Signing XML is also difficult and degrades performance

XML is plaintext. Most implementers have simply begun to encrypt only specific sections of the XML documents to remove a complete loss of performance while gaining as much security as possible. Areas in the XML documents that are left in plaintext are considered low to no risk for the information that a hacker might gain from accessing such documents while they are being transmitted over a network.

Information Leakage: SQL errors

- Not once, not twice, but N times
- The exploitation road map to accessing your data...
- The small to medium company go-to-guy

You cannot give end user clients errors from failures in your dynamic SQL. All errors should be trapped and handled accordingly to give the end user as little information as possible. Does this make help desk much more difficult? Yes. Does this make SQL injections insanely difficult to use (if they exist) to exploit your company? Yes. Information security means keeping information secure.

Information Leakage: Web errors

- Programmers are logical
- Hackers are logical
- Login example
 - Password Invalid
 - User Invalid
 - User or Password Invalid
- Enumeration functions

Web errors give information that allows hackers to find more information about a situation. Any lookup functions on a web site are basically enumeration engines for hackers to query.

Information Leakage: Cookies

- Stored on client
- Modifiable
- Extends to any client side persisted state information
- Serialization
- Client to server program configuration files (non-HTTP)

Cookies are great for persisting state information on the client. And thank goodness that no one every encrypts their contents... I think I'd like another
<delete cookie> <refresh>

Information Leakage: URLs

- URLs tell a story
 - System Administrator/Deployment Know-How
 - Incrementing variables
 - Arguments to functions

They say that a picture is worth a thousand words... and a URL is worth a thousand exposures. Amazing information about a company/system administrator can be found while making queries from a web server.

What Hackers Like

- Information Leakage
 - View state
 - XML
 - SQL errors
 - Web errors
 - Cookies
 - URLs
- Does easy to develop mean easy to exploit?
- Cross Site Scripting
- Replay/Hijacking
- Injection XML/SQL

What information does your application give a user? How can that be used against you? Small to medium businesses suffer due to time-to-market requirements. Are hackers going after the little people more often due to these circumstances?

Information Leakage: Easy Development leads to Easy Exploits

- If I do not incorporate security knowledge and processing during development and deployment of all resources, regardless of whether the access to that resource is anonymous or authenticated, is exploitation possible? YES.

This is the problem with anyone that does not budget for security. Development tools are created to ease development. Not to ease security. Security is a service of a corporation. By performing their due diligence and keeping their customers' needs the most important goal. The quality of a such a product will be sufficient to ward off the hackers of today. Small to medium companies, generally by budget, do not have the resources to perform the detailed security auditing of their own networks and software required to defend against the masses of hackers that exist in the world's population today.

What Hackers Like

- Information Leakage
 - View state
 - XML
 - SQL errors
 - Web errors
 - Cookies
 - URLs
- Does easy to develop mean easy to exploit?
- Cross Site Scripting
- Replay/Hijacking
- Injection XML/SQL

What information does your application give a user? How can that be used against you? Small to medium businesses suffer due to time-to-market requirements. Are hackers going after the little people more often due to these circumstances?

Cross Site Scripting

- HTML inputs for everyone
- How do I validate?
- Just don't do it if you can avoid it... good design makes for good security

Cross Site Scripting is the ability for a user to enter HTML or scripting into an input field and have that HTML or scripting be downloaded by another user, where it executes. This can be used to attack users in many different ways. The scariest part is that it appears that your web server is the culprit. To a public that is unaware of the technological circumstances, it could seem that YOU are attacking them instead of the wily hacker next door who planted the data on your web server.

What Hackers Like

- Information Leakage
 - View state
 - XML
 - SQL errors
 - Web errors
 - Cookies
 - URLs
- Does easy to develop mean easy to exploit?
- Cross Site Scripting
- **Replay/Hijacking**
- Injection XML/SQL

What information does your application give a user? How can that be used against you? Small to medium businesses suffer due to time-to-market requirements. Are hackers going after the little people more often due to these circumstances?

Replay / Hijacking

- Session Hijacking
 - HTTP Session IDs
 - .NET Forms Authentication
- Got SSL?
 - Hey! Cross Site Scripting to the rescue...
- Validation = (Authentication -> Session)
 - * Each Request

Unless you do more than the default amount of session processing, you will always fall prey to the fact that hackers can hijack the sessions of your users... yes even if you use SSL.

What Hackers Like

- Information Leakage
 - View state
 - XML
 - SQL errors
 - Web errors
 - Cookies
 - URLs
- Does easy to develop mean easy to exploit?
- Replay/Hijacking
- Injection XML/SQL

What information does your application give a user? How can that be used against you? Small to medium businesses suffer due to time-to-market requirements. Are hackers going after the little people more often due to these circumstances?

Injection XML/SQL

- SOAP
- Dynamic SQL
- .NET SqlParameter

Stop injection into fields by binding input and avoiding all dynamic SQL in code behind modules. Authenticate and sign all XML data transfers.

.NET from the Hacker's Perspective

What Hackers Dislike
Risk
What Hackers Like
Summary

This presentation entitled “.NET from a hacker’s perspective” focuses on specific .NET technologies on a high-level to show how hackers are approaching this new technology.

Summary

- Parameter validation still key to a majority of vulnerabilities
- Why authenticate when you can hijack?
- Sign code, encrypt data, or else...
- Server side security much better... communication security still difficult to secure with ease, but definitely possible

.NET answers many of the needs of security today. Specifically server side code and data security. With use of SqlParameter objects and store procedures, SQL Injections can be completely removed as a potential vulnerability. With good design and input checking all buffer overflows and CSS attacks can also be mitigated. Adding session security will decrease performance quite a bit, especially when dealing with encrypting view states and other data which requires such security. Cryptography is insanely easy to use, which makes even the most paranoid user feel quite a bit better about their personal data being stored remotely. All things taken into consideration, one can force hackers to go find someone else to hack, if they take the time and money to work on solving these problems in their application. .NET is a step in the right direction. Calculate your risk and act accordingly.

.NET from the Hacker's Perspective

Drew Miller

Drew.miller@securityage.com