Parte V

Segurança de Aplicações na Web

Entrada Inesperada de Dados

Introdução

- A Internet é composta de muitas aplicações, cada qual realizando seu papel.
- Para que uma aplicação útil, ela precisa interagir com um usuário.

Introdução

- Aplicações:
  - cliente de chat;
  - site de e-commerce;
  - utilitário de sistema para linha de comando;
  - um jogo on-line;
  - uma aplicação de cáculo;
  - ... ...

Introdução

- Todas as aplicações modificam sua execução dinamicamente com base na entrada do usuário.

- Estar na Internet significa que a aplicação pode ser acessada remotamente por vários usuários.

Introdução

- Se for mal codificada, a aplicação poderá deixar seu sistema aberto a vulnerabilidades quanto a sua segurança.

- Uma codificação fraca pode ser o resultado da falta de experiência, um erro de programação ou uma anomalia não considerada.

Introdução

- Grandes aplicações normalmente são desenvolvidas em partes menores e reunidas para gerar um projeto final.

- É possível que existam diferenças nas partes que, quando combinadas com outras partes, resultem em vulnerabilidades.

Introdução

- Desenvolvedores de aplicações versus administradores de rede
- Desenvolvedores normalmente não são conscientes de segurança.
- Se não existir uma política de segurança documentada, indicando a segurança como requisito para a aplicação, é difícil fazer com que os desenvolvedores não conscientes tornem as aplicações seguras.

Introdução

- A proliferação de vulnerabilidades devido a dados de entrada inesperados é muito alta.

Dados Inesperados

- Para interagir com um usuário, uma aplicação precisa aceitar dados fornecidos pelo usuário.

- Dados podem estar em formato simples (clique do mouse ou único caractere) ou em um fluxo complexo (quantidades consideráveis de dados).

Dados Inesperados

- O usuário pode, sabendo ou não, submeter dados que a aplicação não está esperando.

- O resultado pode ser nulo ou então modificar a resposta programada pela aplicação.

Dados Inesperados

- Isso pode levar a aplicação a oferecer informações aos usuários que eles normalmente não podem obter, ou então violar a aplicação ou até o sistema operacional.

Ataques

- Dados inesperados podem resultar em três tipos de ataques:

  - Buffer Overflow
  - Funções do Sistema
  - Alteração da lógica

Ataques

- Buffer Overflow – quando o atacante submete mais dados do que a aplicação espera. A aplicação pode não lidar de forma correta com os dados excedentes.

Ataques

- Buffer Overflow

  C e C++ não lidam corretamente com dados excedentes.

  A aplicação precisa ser programada especificamente para lidar com eles.

Ataques

- Buffer Overflow

  Perl e PHP tratam automaticamente os dados excedentes, aumentando o tamanho do espaço para armazenamento variável.

Ataques

- Funções do sistema – Os dados são usados de alguma forma para interagir com um recurso que não está contido dentro da própria aplicação.
- As funções do sistema incluem, o acesso, o trabalho com arquivos ou a execução de outras aplicações. Os dados também podem modificar o comportamento de uma função do sistema.

Ataques

- Alteração da lógica – Os dados podem ser preparados de tal forma que modifiquem o modo como a lógica da aplicação os deve tratar.

Ataques

- Exemplos: desvio de mecanismo de autenticação, a alteração das consultas em SQL e a obtenção de acesso a partes da aplicação às quais o atacante normalmente não teria acesso.

Ataques

- Ataques, em particular, podem ser enquadrados em mais de uma classe de ataque.
- O formato real dos dados inesperados pode ser o nome de um arquivo alternativo ou a inclusão de metacaracteres especiais, que possuem significado alternativo para a aplicação ou para o sistema que a executa.

Situações com Dados Inesperados

- Aplicações e utilitários locais;
- Dados Inesperados no HTTP/HTML;
- Dados Inesperados nas Consultas SQL;
- Autenticação da aplicação;
- Disfarçando do IDS (Intrusion Detection System);

Situações com Dados Inesperados

- Aplicações e utilitários locais

A aplicação interage com o usuário e dá a um usuário malicioso a chance de fazer que a aplicação não espera.

Isso pode ser: digitar uma sequência de teclas anormal, a inclusão de uma grande quantidade de dados ou a especificação dos tipos errados de valores.

Situações com Dados Inesperados

- Aplicações e utilitários locais

  Isso não é um grande problema, se o usuário fizer algo errado a aplicação falhará.

Situações com Dados Inesperados

- Aplicações e utilitários locais

- No mundo UNIX algumas aplicações possuem permissões especiais chamadas *set user ID* (*suid*) e *set group ID* (*sgid*).

Situações com Dados Inesperados

- Aplicações e utilitários locais
- Enganar uma aplicação *suid* ou *sgid* pode resultar no privilégio dos administradores.

Técnicas para Localizar e Eliminar Vulnerabilidades

- Suas próprias aplicações são vulneráveis?
- Como se pode descobrir?
- Algumas técnicas comuns para determinar se uma aplicação é vulnerável, e se for, consertá-la.

Técnicas para Localizar e Eliminar Vulnerabilidades

- Teste da Caixa-Preta;
- Uso de Código-Fonte;
- Filtragem Apropriada e Escape de Caracteres;
- Remoção silenciosa de dados, funções de filtragem centralizada.

Teste da Caixa-Preta

- Aplicações Web
- Interesse especial em formulários FHTM e URL's com parâmetros.
- Parâmetros são valores após o símbolo "?" na URL.
- Localizar uma aplicação Web que contenha páginas dinâmicas com muitos parâmetros na URL.

Teste da Caixa-Preta

- Para começar, altere alguns dos valores ... ...
- Use a intuição sobre o que a aplicação está fazendo.

- Passar pelo processo interativo completo, do início ao fim, pelo menos uma vez.

Teste da Caixa-Preta

- Tente causar um erro intencional.
- Trabalhe metodicamente para cada parêmetro, inserindo um apóstrofo (‘) depois, aspas (“).
- Tente determinar a necessidade e/ou a utilidade de cada parâmetro.

Teste da Caixa-Preta

- Leve me consideração a postura geral apresentada pelo *site* e pela aplicação, usando isso para deduzir possíveis aspectos da aplicação.

Teste da Caixa-Preta

- Procure qualquer sinal de algo que se pareça com um nome de arquivo.

- Pesquise e entenda as limitações tecnológicas dos diferentes tipos de servidores Web, linguagens de *scripting* ou de aplicação e servidores de bancos de dados.

Teste da Caixa-Preta

- Procure qualquer coisa que e pareça com uma equação, fórmula ou trechos reais de código de programação.

- Coloque-se na posição de um codificador.

Táticas para os Serviços de Rede

- O mundo não é composto meramente de aplicações Web, assim, algumas táticas para verificar vulnerabilidades nos serviços de rede devem ser utilizadas.

Problemas Locais a um Sistema

- Ao examinar os utilitários *suid* / *sgid* faça o seguinte:

- ... .

Usar o Código-Fonte

- A auditoria da aplicação é muito mais eficaz se você tiver o código-fonte da aplicação que deseja explorar.

- Pode-se usar técnicas como *diffing* para localizar vulnerabilidades ou mudanças entre versões.

Usar o Código-Fonte

- Como descobrir uma situação onde a aplicação pode ser explorada por dados inesperados?

Usar o Código-Fonte

- Resposta:

  Procurar chamadas às funções do sistema e rastrear até o local onde os dados foram introduzidos na função.

  Se os dados se originam do usuário, deve-se examinar mais para se determinar se isso pode ser explorado.

  ... ...

Usar o Código-Fonte

Rastreando para frente a partir do ponto de entrada de dados, poderemos chegar a becos sem saída.

Começando nas funções do sistema e voltando, poderemos acompanhar a aplicação no sentido inverso a sua execução.

Usar o Código-Fonte

- Quais funções deveremos procurar ?

  Resposta: depende da linguagem sendo utilizada para construir a aplicação.

- Para operações com arquivos: *open*, *fopen*, ... Para execução de programas: exec, system, ... As consultas de bancos de dados em SQL: *select* ...

Usar o Código-Fonte

- O ideal é que sejam rastreados todos os dados recebidos do usuário, para determinar cada local em que os dados são usados.

- A partir daí, poder-se-á determinar se os dados do usuário terão chance para fazer algo inesperado.

Filtragem Apropriada

- A melhor maneira de combater dados inesperados é filtrar os dados para os valores esperados.

- Avalie quais caracteres são necessários para cada item que o usuário pode enviar.

Filtragem Apropriada

- Observe que podemos ser tolerantes e permitir caracteres de formatação, mas para cada caractere permitido, estaremos aumentando o risco em potencial.

Escape de Caracteres

- Existem caracteres (por exemplo, uma contrabarra) que possuem significado especial para outros ambientes, como por exemplo, para os *shells* de linha de comando do UNIX.

Escape de Caracteres

- Então, é preciso que se tenha cuidado com esses caracteres, quando colocados como entrada de dados por um usuário.

- A idéia inicial é “escapar” desses caracteres usando caracteres de escape, como por exemplo, usar uma contrabarra (\) para escapar um retorno de carro ou escapar um caractere NULL.

Escape de Caracteres

- Mas, por que escapá-los, se sequer precisamos deles?

- Existem casos onde os caracteres de escape (por exemplo, contrabarra) nem sempre são suficientes.

Escape de Caracteres

- O importante é remover os dados problemáticos, ... , removendo a dúvida.
- Cada linguagem possui seu modo próprio, através de funções nativas, de filtrar e remover caracteres dos dados de entrada do usuário.
- Perl, ColdFusion ou CFML, ASP, PHP, SQL

Remoção Silenciosa ou Alerta

- Ao lidar com dados recebidos do usuário, temos duas opções:
  1. remover os caracteres com problema, manter os caracteres corretos e continuar processando com o que restou;
  2. parar imediatamente e alertar a entrada inválida.

Remoção Silenciosa ou Alerta

- Cada técnica possui pós e contras.
- Se a aplicação alerta o usuário que ele submeteu dados errados, isso permite a um atacante ver quais caracteres a aplicação está procurando ...

Remoção Silenciosa ou Alerta

- Essa técnica é útil para se determinar as vulnerabilidades nas aplicações quando não se tem acesso ao código-fonte.

- A filtragem silenciosa dos dados para incluir apenas os caracteres seguros gera alguns problemas diferentes:

Remoção Silenciosa ou Alerta

1. Os dados estão sendo alterados, e se a integridade dos dados submetidos tem que ser exata (por exemplo, em senhas)
... ...

2. Se a aplicação imprimir os dados submetidos, depois de filtrados, um atacante poderá ver o que está sendo removido na submissão de dados.

Remoção Silenciosa ou Alerta

- A solução apropriada depende da aplicação.
- Recomenda-se a combinação das duas técnicas, dependendo do tipo e da integridade necessária para cada dado submetido.

Função de Entrada Inválida

- A centralização de uma função para ser usada para informar dados inválidos.

- Saber se os usuários estão realmente tentando submeter caracteres (indesejados) que a aplicação filtra é importante.

Função de Entrada Inválida

- O desenvolvedor saberá quando e como um invasor está tentando subverter a lógica da sua aplicação.

- A função serve para monitorar as violações (colocar os dados inesperados num arquivo de *log*) e determinar se ocorreu uma violação ou se foi um erro casual.

Função de Entrada Inválida

- Com a coleta dessa informação pode-se ter uma análise estatística, para se determinar que tipo de caracteres espera-se receber, e assim, poder-se ajustar os filtros com mais precisão.

Função de Entrada Inválida

- Pode-se também usar a função de violação para imprimir um alerta de entrada inválida e abortar a aplicação.

Medidas de Segurança nas Linguagens

- Muitas linguagens de programação e aplicações já possuem recursos que lhe permitem reduzir os riscos dos dados adulterados.

- Os recursos podem manter os dados adulterados em quarentena até que eles sejam revisados.

Medidas de Segurança nas Linguagens

- Recursos de PHP
- Opção de configuração em “safe mode” que limita o usos de funções do sistema em PHP.
- Não ajuda diretamente a limpar os dados recebidos do usuário;
- Serve como proteção, caso um invasor encontre um meio de evitar suas verificações de adulteração.

Medidas de Segurança nas Linguagens

- PHP
- Quando o modo seguro está ativado, PHP limita as seguintes funções para que somente possam acessar arquivos pertencentes à ID do usuário (UID) da PHP (normalmente é a UID do servidor Web), ou arquivos em um diretório possuidos pela UID da PHP:

Medidas de Segurança nas Linguagens

- PHP
- Funções: *include*, *readfile*, *fopen*, *file*, *link*, *unlink*, *symlink*, *rename*, *rmdir*, *chmod*, *chown* e *chgrp*.
- Limita o uso de *exec*, *system*, *passthru* e *popen* para somente executar as aplicações contidas no diretório **PHP_SAFE_MODE_EXEC_DIR**, definida em php.h quando PHP é compilada.

Medidas de Segurança nas Linguagens

- PHP
- *Mysql_Connect* está limitado a só permitir conexões de banco de dados com a UID do servidor Web ou a UID do *script* em execução para a aplicação.

Medidas de Segurança nas Linguagens

- PHP

- Modifica o modo como a autenticação baseada em HTTP é tratada, a fim de evitar truques enganosos (que é o maior problema de sistemas que contém muitos *sites* hospedados).

Medidas de Segurança nas Linguagens

- ASP
- VBscript e JScript
- Não contém muitas funções relacionadas ao sistema.
- O que está disponível são as funções do sistema de arquivos (padrão).
- Contém uma chave de configuração (configuration key) que impede a notação “../” ser usada nas funções do sistema de arquivos.

Medidas de Segurança nas Linguagens

- ASP
- Assim, limita a possibilidade de um invasor obter acesso a um arquivo fora do diretório raiz da Web.
- Pode desabilitar os *Parent Paths* "caminhos ao pai" com o Microsoft Management Console (console de configuração do IIS), assim:

Medidas de Segurança nas Linguagens

- ASP
  - Selecione o *site* de destino |
  - Properties |
  - Home Directory |
  - Configuration |
  - Application Options |
  - Desabilita Enable Parent Paths

Medidas de Segurança nas Linguagens

- ASP
- Se não precisa do suporte do sistema de arquivos em seus documentos ASP, pode-se removê-lo, retirando seu registro do File System Object; para isso execute no prompt:

  ...> regsvr32 scrrun.dll /u

Medidas de Segurança nas Linguagens

- MySQL
- Possui a capacidade de, durante as consultas, ler dados de um arquivo e armazenar no banco ou consultar no banco e armazenar em arquivo, usando:

```
SELECT  *  INTO OUTFILE  “/temp/save.txt”  FROM  table
```

Pode-se limitar esse comportamento não concedendo permissões para armazenar em arquivos, a quaisquer usuários na tabela de privilégios embutida no MySQL.

Ferramentas

- Lidam com a entrada de dados inesperados.
- Algumas úteis para programadores consertarem seu código.
- Outras úteis para atacantes ou consultores, procurando vulnerabilidades.

Ferramentas

- Web Sleuth
  - Windows
  - o usuário pode modificar diversos aspectos dos pedidos HTTP e formulários HTML;
  - Extensível através de plug-ins;
  - Rastreamento de sites Web;
  - Testes de consultas SQL;
  - Gratuita em www.owasp.org

Ferramentas

- RATS
  - Rough Auditing Tool for Security;
  - Revisão de código-fonte que entende C, C++, Python, Perl e PHP;
  - Alerta para quaisquer situações potencionalmente perigosas, como buffers estáticos e funções sem segurança;
  - Ajuda a reduzir vulnerabilidades;
  - Gratuita em www.securesw.com

Ferramentas

- Flawfinder
  - Script Python similar a RATS;
  - Limitado ao código em C;
  - O objetivo é integrar com RATS;
  - Gratuita em www.dhwheeler.com

Ferramentas

- Retina
  - Analisador comercial de vulnerabilidades;
  - Permite ao usuário analisar novas vulnerabilidades nas aplicações;
  - Possui CHAM (Common Hacking Attack Methods), que automatiza a procura de *buffer overflow* e problemas nos serviços acessíveis pela rede;
  - www.eeye.com (eEye)

Ferramentas

- Hailstorm
  - Injeção de Falhas;
  - Tem mais recursos do que o CHAM de Retina;

  - Um conjunto de ferramentas e um motor de scripting interno em Perl que permite que se crie os tipos de testes de anomalia, para serem lançados contra a aplicação.

  - Exige conhecimento na arte de descobrir bugs;

  - Disponível comercialmente em www.clicktosecure.com

Ferramentas

- Pudding
  - Um proxy HTTP escrito em Perl.
  - A finalidade é evitar os IDS (Intrusion Detection Systems);
  - Inclui vários truques de codificação de URL's HTTP para quaisquer pedidos que passe nela;
  - Pedidos originados no browser ou ferramenta de avaliação da Web;
  - Codificação mais popular: UTF-8/Unicode;
  - Gratuita em www.packetstormsecurity.org

Ferramentas

- CGIAudit
  - Ferramenta de caixa-preta CGI automatizada;
  - Toma uma definição de formulário HTML fornecida pelo usuário;
  - Testa cada elemento do formulário, buscando vulnerabilidades como, *buffer overflow*, execução de meta-caracteres e modificação de consultas SQL;
  - Possui um *Web spider* e suporte para *proxy HTTP*;
  - Escrita em C;
  - Disponível em www.innu.org/~super

Situações com Dados Inesperados