Todos atentos à lista de chamadas do professor Brute

mConnect MV - Descobrindo usuários válidos

Iran Macedo
Hacker / Pentester / Segurança ofensiva
Published May 9, 2020

Olá!

Continuando no assunto de Brute Force, abordarei um dos métodos que podemos utilizar para descobrir os usuários das credenciais de uma aplicação falha na sua segurança, que também serve de base para o CVE-2020-23283. Esta aplicação é real e descontinuada pelo seu desenvolvedor. Os usuários são reais e estão realmente cadastrados na aplicação.

Aproveito para informar que a ação de coletar informações de serviços não é caracterizado crime, uma vez que os serviços estão em um ambiente controlado de testes. Esta matéria tem como fins apresentar os métodos utilizados por hackers para encontrar informações de empresas e apresentar aos desenvolvedores as falhas de segurança que suas aplicações web (portais, sites de atendimento, etc) possuem. Corrigir ou não é uma decisão dos desenvolvedores e gestores/clientes destas aplicações.

Por Iran Macedo, especialista em proteção de dados e segurança ofensiva / Pentest. 09/05/2020.

Quebrar credencial é muito mais do que apenas conhecer uma senha

A primeira informação que precisamos entender é que uma credencial de acesso, geralmente, é formada por um usuário e por uma senha. Desta forma, se eu entender isso e se eu for um cara cuidadoso, farei com que o meu usuário seja tão difícil de adivinhar quanto a senha que uso nos serviços online.

Lógico que, por muitas vezes, isso é difícil, já que alguns serviços, como o e-mail, precisamos passar para outras pessoas como meio de contato e este tipo de informação se torna pública.

Porém, o mesmo não precisa ser assim em outros serviços, ainda mais nas aplicações privadas de empresas. E este é o caso, onde vamos explorar uma falha da aplicação para encontrar a primeira parte de uma credencial de login: um usuário válido.

Como eu disse na primeira matéria sobre Brute Force, não basta apenas ter várias chaves que encontrei nas mãos se eu não conhecer quais portas estas chaves abrem.

A aplicação falha

desenvolvedor e não são mais ofertadas aos seus clientes, cujas versões mais novas e livres destas falhas encontram-se disponíveis para atualização e uso.

Entramos em contato com o desenvolvedor VivaceMV (www.mv.com.br) para checar primeiro se as versões encontravam-se desatualizadas para, se não, reportar primeiramente ao desenvolvedor sobre as falhas encontradas.

Foram encontradas falhas severas de tratamento básico, o que torna um enorme risco a disponibilização destas aplicações na internet. Foram levantadas falhas de:

- Leak de dados que expõem os usuários cadastrados através da página de login;
- Leak de outras várias informações gravadas no banco de dados através do código-fonte;
- Falha de SQL Injection através da página de login.

A falha

A técnica que vou utilizar para explorar esta aplicação e encontrar usuários válidos se dá pela falha na saída da informação de erro de login, como também pela falta de um controle e bloqueio inteligente de entrada repetidas de dados.

O que isso significa?

Significa que a própria aplicação me diz se um usuário existe ou não. E significa também que eu posso disparar quantos nomes eu quiser, pois nem a aplicação, nem o servidor onde ela está hospedada possuem um controle de tentativas e erros que possam parar este ataque de força bruta.

Vejamos... vou inserir um nome qualquer, que escolhi a esmo: amanda. O que a aplicação me retorna: Senha inválida.

Ok. Eu não digitei senha alguma, somente o nome e enviei o pedido de acesso assim. Usuário: amanda, Senha: nula. O retorno que eu tive da aplicação foi que a senha está errada.

Entendo então que o usuário existe, já que o retorno não foi de usuário inválido. Será?

O que acontece então se eu colocar um outro nome que não exista na aplicação. Que tal o nome: amando. Vejamos...

Agora o retorno que tenho é que o usuário não existe, pois não foi encontrado no banco de dados da aplicação. Veja que a aplicação, que é falha no tratamento das suas informações de saída de erro, me informa gentilmente quais usuários existem no seu banco. Para os que existem, a aplicação informa desconhecer a senha em branco enviada.

O ataque

O ataque é simples e bem básico. Primeiro de tudo, preciso de uma grande lista com muitos nomes. Eu posso utilizar o próprio site da empresa e informações da mídia para descobrir nomes dos colaboradores e utilizá-los na exploração da falha. Muitas empresas utilizam nome simples (apenas o nome), outras utilizam nomes compostos (nome e

Como eu descobri que esta aplicação trabalha com nomes simples, vou focar somente neste tipo de login, mesmo que ela aceite nomes compostos. E, sim, ela aceita também nomes compostos, pois eu também encontrei estes tipos de usuários nesta mesma aplicação.

Para não ter de escrever um monte de nomes, pesquisei na internet e encontrei uma lista com 20.000 nomes brasileiros, femininos e masculinos, para baixar como arquivo CSV.

Como os nomes vem em uma lista que possui outras informações, é necessário filtrá-la para um arquivo que tenha somente os nomes, ignorando todas as demais informações.

Vamos dar alguns comandos no Bash (cut) para limpar essa sopa de caracteres aí.

Agora sim temos uma lista com 20.000 nomes para utilizar na descoberta de logins.

A ferramenta de Brute Force

Tentei gerar um script em Shell para atacar esta aplicação, mas ela trabalha na linguagem ASPX e o Curl não lida bem com este tipo de extensão, pois o Curl apresenta na tela o código binário e não o texto do HTML da página.

Desta forma e para não gastar muito tempo, vou utilizar uma das ferramentas mais utilizadas para exploração de aplicações web no mundo Hacker: o Burp Suite.

Acertando o navegador para utilizar o Burp como Proxy, faço um acesso para verificar o retorno da aplicação.

Agora, pegando a tarefa acima (ID 148), com o botão direito eu a envio para o Intruder do Burp. O Intruder é o cara de Brute Force do Burp Suite.

No Intruder eu configuro todos os itens necessários, começando pelo alvo (endereço da aplicação) e porta de conexão da aplicação.

Na guia 'Positions' nós configuramos os campos que utilizaremos no ataque. O tipo de ataque será o Sniper.

A primeira coisa é clicar no botão 'Clear' para limpar a seleção prévia de todos os campos, selecionar o nome que você utilizou para encontrar um usuário válido e, depois do nome selecionado (no exemplo, amanda), clicar no botão 'Add'. Assim informamos ao Burp que iremos trabalhar somente com este campo, o Username.

Na aba 'Payloads' vamos utilizar o tipo de lista simples (Simple list) e clicando sobre o botão 'Load', vamos selecionar a lista com 20.000 nomes que criamos.

Para facilitar a identificação de nomes válidos de nomes inválidos, precisamos criar um filtro. Isso jogará na tela uma mensagem visual que nos facilitará identificar o que deu certo do que deu errado. Sem isso a verificação final precisaria ser manual. 20.000 tentativas, manualmente, uma a uma.

Para isso vou utilizar a guia 'Options' e clicar sobre o botão 'Add'.

Vou selecionar a informação que eu quero filtrar, que é "Senha inválida". Automaticamente a página se encarregará de completar os dados de limitações de campo de pesquisa para você (:Red;"> e).

E o filtro ficará desta forma. Só precisa tomar cuidado para verificar se estes mesmos limites não existem também em outros pontos do HTML da página.

Após tudo configurado, basta clicar no botão 'Start attack' e aguardar o Burp esgotar todos os 20.000 nomes (!) da lista.

Para organizar a saída, basta clicar no cabeçalho da coluna do filtro que criamos (:Red;">) para separar os usuários encontrados (Senha inválida) dos usuários que não existem no banco de dados da aplicação (Usuário não encontrado).

Ao final, tendo em mãos todos os usuários que existem, basta exportar os resultados, tratar os dados para excluir os usuários inválidos e começar um novo ataque Brute Force, agora testando uma lista de senhas sobre os usuários existentes da aplicação.

Como a aplicação é bem falha no tratamento das informações, é bem capaz dela aceitar senhas fáceis e simples de adivinhar, como '1234' ou 'abcd', etc.

Agora ficou mais fácil seguir com a coleta de senhas válidas, pois vamos economizar esforços ao não disparar milhões de senhas contra usuários que não existem.

Soluções para esta falha de segurança

Primeiro de tudo, uma aplicação nunca deve informar ao usuário se o seu login digitado ou se a sua senha digitada estão incorretos. A mensagem, como já estamos habituados a ver em várias outras aplicações web, deve ser genérica e não passar com exatidão o que está errado. Uma mensagem de erro do tipo "Seus dados estão incorretos" ou "Usuário ou senha não confere" já seria o suficiente para que eu não soubesse quais dos dois dados passados estão errados. O nome desta técnica é "Proteção por obscuridade", ou seja, informar desinformando, criando dúvidas. E, convenhamos, é muito mais fácil eu errar uma senha do que um usuário. Portanto, eu ficaria sem saber se o usuário informado realmente existe ou não, pela possibilidade enorme de ter errado a senha.

Segundo, aplicações web devem possuir o mínimo de proteção contra ataques de força bruta. Poderia ser algum mecanismo da própria aplicação, solicitando uma comprovação do usuário de que o mesmo não é um robô através de Captchas, daí passando para métodos de proteção mais avançados, mais confiáveis e mais caros, como bloqueios no firewall do servidor por um IPS ou utilizando um firewall web (Web Application Firewall ou WAF) para uma análise prévia das requisições de logon na aplicação.

Correção da falha

Segundo informações passadas pelo desenvolvedor, o programa mConnect foi descontinuado e não recebe mais atualizações ou suporte. Para continuar a utilizar o serviço, o novo programa, que substituiu o mConnect é o Vivace Connect, disponível para aquisição através de contato com o desenvolvedor.

Conclusão

A sua empresa pode ser severamente afetada por um ataque hacker que só foi possível porque a aplicação utilizada, que você comprou com uma empresa desenvolvedora, não é segura e possui falhas de segurança.

Nem toda falha joga o hacker diretamente dentro da aplicação ou do servidor. Mas pode ser utilizada para que o hacker descubra os dados necessários para entrar nesta aplicação e ter acesso às informações internas da sua empresa.

Desta forma, entendemos que é da responsabilidade do desenvolvedor corrigir estas falhas. Mas, sendo cauteloso, indico que as empresas que utilizam aplicações de terceiros, testem também tais aplicações à procura de falhas, antes de disponibilizar seus acessos e dados privados na internet.

Afinal, quem é que pagará o preço por ter dados acessados, expostos, alterados, sequestrados ou apagados? O cliente que comprou a aplicação ou o desenvolvedor da aplicação?

Olha aí, LGPD! Um sério candidato à multas por vazamento de informações de pacientes. O que a LGPD diz sobre isso?

Isso é papo para ser levado ao juízo, não é mesmo? Afinal, todos têm a sua parcela de culpa.

Abraço!

Documentação formal para o CVE na Mitre: MeuGithub

CVE ID: CVE-2020-23283

Interviewing

Salary and Compensation

Internships

Employee Benefits

See All