

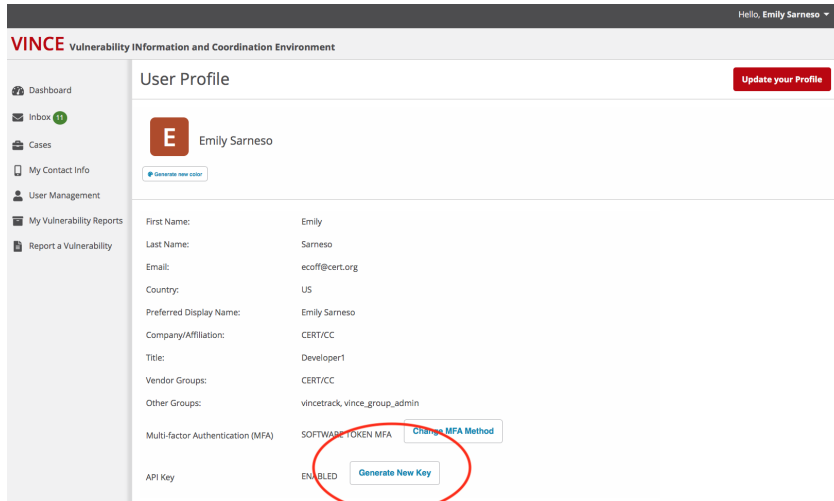
VINCE API

The VINCE API is a work in progress. Please provide [feedback](#) through VINCE or [GitHub](#).

- [Authentication](#)
- [Code Examples](#)
 - [List cases you are participating in](#)
 - [Get case metadata](#)
 - [Get message posts for case](#)
 - [Get original report for case](#)
 - [List vulnerabilities for case](#)
 - [List vendors for case](#)
 - [List vendors including status and statement for each vulnerability](#)
 - [Get Vulnerability Note text \(if it exists\)](#)
 - [Get Vulnerability Advisory in CSAF format](#)
 - [Update vendor status](#)
 - [Look up CVE IDs \(must have access to case\)](#)

Authentication

1. Log in to [VINCE](#).
2. Go to your [User Profile](#).
3. Scroll down to "Generate API Key".
4. Copy the API key to a safe place, you will not be able to access it again. If lost, you need to regenerate a new one.
5. Use the API key in the headers of your request as shown below.



The screenshot shows the 'User Profile' page for Emily Sarneso. The page includes a sidebar with navigation options like 'Dashboard', 'Inbox', 'Cases', 'My Contact Info', 'User Management', 'My Vulnerability Reports', and 'Report a Vulnerability'. The main content area displays user details such as 'First Name: Emily', 'Last Name: Sarneso', 'Email: ecoff@cert.org', 'Country: US', 'Preferred Display Name: Emily Sarneso', 'Company/Affiliation: CERT/CC', 'Title: Developer1', 'Vendor Groups: CERT/CC', and 'Other Groups: vincetrack, vince_group_admin'. At the bottom, the 'Multi-factor Authentication (MFA)' is set to 'SOFTWARE TOKEN MFA' with a 'Change MFA Method' button. The 'API Key' is 'ENABLED' with a 'Generate New Key' button circled in red.

```
headers={'Authorization': "Token {}".format(token)}
```

Code Examples

List groups (organizations) you belong to

```
# get information about organizations you belong to:  
api = 'https://kb.cert.org/vince/comm/api/vendor/'  
headers={'Authorization': "Token {}".format(token) }  
r = requests.get(api, headers=headers, stream=True)  
print(r.text)
```

```

API: /vince/comm/api/vendor #get information about vendors you belong to
[  {  'emails': ['test@example.com'],
      'id': 3548,
      'users': ['vince.user'],
      'vendor_name': 'VendorCorp'},
    {  'emails': ['test@example.com'],
      'id': 3551,
      'users': ['vince.user'],
      'vendor_name': 'Testing Co'},
    {  'emails': ['test@example.com', 'test3@example.com'],
      'id': 3549,
      'users': ['vince.user', 'Vince User'],
      'vendor_name': 'Testing Vendor'}]

```

List cases you are participating in

```

# get a list of your cases
headers={'Authorization': "Token {}".format(token)}
api = 'https://kb.cert.org/vince/comm/api/cases/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)

```

```

API: /vince/comm/api/cases # get a list of cases involved in
[  {  'created': '2020-06-11T18:51:48.204903Z',
      'due_date': None,
      'status': 'Active',
      'summary': 'test',
      'title': 'test',
      'vuid': '782161'},
    {  'created': '2020-04-28T19:48:50.216317Z',
      'due_date': '2018-07-23T14:20:09Z',
      'status': 'Inactive',
      'summary': 'TechSmash firmware or operating system software drivers '
                  'may not sufficiently validate elliptic curve parameters '
                  'used to generate public keys during a Diffie-Hellman key '
                  'exchange, which may allow a remote attacker to obtain the '
                  'encryption key used by the device',
      'title': 'TechSmash implementations may not sufficiently validate '
                'elliptic curve parameters during Diffie-Hellman key exchange',
      'vuid': '3123125'}}]

```

Get case metadata

```

# get information about VU#701852
api = 'https://kb.cert.org/vince/comm/api/case/701852/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)

```

```

API: vince/comm/api/case/701852/ # get information about a specific case
{  'created': '2020-06-11T18:51:48.204903Z',
    'due_date': None,
    'status': 'Active',
    'summary': 'test',
    'title': 'test',
    'vuid': '785701'}

```

Get message posts for case

```
# get all posts for case VU#701852
api = 'https://kb.cert.org/vince/comm/api/case/701852/posts/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
API: vince/comm/api/case/701852/posts/ # get all posts for a specific case
[ { 'author': 'vince.user',
  'content': 'The [draft vulnerability '
            'note](http://localhost:8000/vince/comm/case/18/notedraft/) '
            'has been updated.',
  'created': '2020-11-17T19:13:07.866230Z',
  'pinned': True},
  { 'author': 'vince.user',
  'content': 'Please [view this draft vulnerability '
            'note](http://localhost:8000/vince/comm/case/18/notedraft/).',
  'created': '2020-11-17T19:07:56.624450Z',
  'pinned': True},
  { 'author': 'vince.user',
  'content': 'test 2',
  'created': '2020-10-29T19:49:33.422875Z',
  'pinned': False},
  { 'author': 'vince.user',
  'content': 'test 1',
  'created': '2020-10-29T19:49:30.434164Z',
  'pinned': False}]
```

Get original report for case

```
# get the original report for VU#701852
api = 'https://kb.cert.org/vince/comm/api/case/701852/report/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
API: /vince/comm/case/701582/report/ # get report for a specific case
{ 'contact_email': 'joebob@vendor.example.com',
  'contact_name': 'Joe Bob',
  'contact_org': 'VendorExample',
  'contact_phone': '5551231234',
  'date_submitted': '2020-06-08T20:01:47.896419Z',
  'disclosure_plans': '',
  'exploit_references': '',
  'product_name': 'test',
  'product_version': 'v. 12.3',
  'public_references': '',
  'share_release': True,
  'vendor_name': 'Test Vendor',
  'vul_description': 'This is the description',
  'vul_disclose': True,
  'vul_discovery': 'This is the discovery.',
  'vul_exploit': 'This is the exploit',
  'vul_exploited': True,
  'vul_impact': 'This is the impact',
  'vul_public': True}
```

List vulnerabilities for case

```
# get the vuls for VU#701852
api = 'https://kb.cert.org/vince/comm/api/case/701852/vuls/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
API: /vince/comm/case/701582/vuls/ # get vuls for a specific case
[ { 'cve': None,
  'date_added': '2020-11-19T21:43:17.210726Z',
  'description': 'This is another vul without a cve.',
  'name': 'VU#785701.2'},
  { 'cve': '2020-19293',
  'date_added': '2020-10-22T15:30:11.888074Z',
  'description': 'Test this is a vul.',
  'name': 'CVE-2020-19293'}]
```

List vendors for case

```
# get all the vendors involved in VU#701582 (also gets their status and statements)
api = 'https://kb.cert.org/vince/comm/api/case/701852/vendors/'
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
API: /vince/comm/case/701582/vendors/ # get vendors for a specific case
[ { 'cert_addendum': None,
  'date_added': '2020-11-20T14:40:24.080886Z',
  'references': 'http://www.example.com\nhttps://www.example.org',
  'statement': 'Test',
  'statement_date': '2020-11-23T19:50:44.813809Z',
  'status': 'Unknown',
  'vendor': 'VendorCorp'},
  { 'cert_addendum': None,
  'date_added': '2020-10-08T18:27:41.526942Z',
  'references': 'http://www.example.com\nhttps://www.example.org',
  'statement': 'Test',
  'statement_date': '2020-11-19T21:26:32.399730Z',
  'status': 'Affected',
  'vendor': 'Testing Co'}]
```

List vendors including status and statement for each vulnerability

```
# get all the vendors and their status/statement/references for each specific vul
api = f'https://kb.cert.org/vince/comm/api/case/701582/vendors/vuls/'
headers={'content-type': 'application/json', 'Authorization': "Token {}".format(token) }
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```

API: /vince/comm/case/701582/vendors/vuls/ # get vendors status for specific vuls
[  {  'references': 'http://www.example.com\nhttps://www.example.org',
    'statement': 'Test',
    'statement_date': '2020-11-19T21:47:44.239683Z',
    'status': 'Affected',
    'vendor': 'Testing Co',
    'vulnerability': 'VU#785701.2'},
  {  'references': 'http://www.example.com\nhttps://www.example.org',
    'statement': 'This is my statement',
    'statement_date': '2020-10-22T15:38:11.859615Z',
    'status': 'Not Affected',
    'vendor': 'Testing Co',
    'vulnerability': 'CVE-2020-19293'},
  {  'references': '',
    'statement': '',
    'statement_date': '2020-11-20T15:23:18.997947Z',
    'status': 'Unknown',
    'vendor': 'VendorCorp',
    'vulnerability': 'VU#785701.2'},
  {  'references': '',
    'statement': '',
    'statement_date': '2020-11-20T15:23:18.938232Z',
    'status': 'Unknown',
    'vendor': 'VendorCorp',
    'vulnerability': 'CVE-2020-19293'}]

```

Get Vulnerability Note text (if it exists)

```

# get the vulnerability note, if available
api = f'https://kb.cert.org/vince/comm/api/case/701582/note'
headers={'content-type': 'application/json', 'Authorization': "Token {}".format(token) }
r = requests.get(api, headers=headers, stream=True)
print(r.text)

```

```

#API: /vince/comm/api/case/710582/note/ # get draft vul note
{  'content': '### Overview\r\n'
    '\r\n'
    'Testing API so need some content.\r\n'
    '\r\n'
    '\r\n'
    '### Description\r\n'
    '\r\n'
    '### Impact\r\n'
    'The complete impact of this vulnerability is not yet known.\r\n'
    '\r\n'
    '### Solution\r\n'
    'The CERT/CC is currently unaware of a practical solution to '
    'this problem.\r\n'
    '\r\n'
    '### Acknowledgements\r\n'
    'Thanks to the reporter who wishes to remain anonymous.\r\n'
    '\r\n'
    'This document was written by Emily Sarneso.',
  'datefirstpublished': None,
  'dateupdated': '2020-11-17T19:13:07.755453Z',
  'published': False,
  'references': ['www.example.org', 'www.example.com'],
  'revision': 2,
  'title': 'test',
  'vuid': '785701'}

```

Get Vulnerability Advisory in CSAF format

```
# get the vulnerability note, if available
api = f'https://kb.cert.org/vince/comm/api/case/495801/csaf/'
headers={'content-type': 'application/json', 'Authorization': "Token {}".format(token) }
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
#API: /vince/comm/api/case/495801/csaf/ # get draft vul note {
  "document": {
    "acknowledgments": [
      {
        "urls": [
          "https://kb.cert.org/vuls/id/495801#acknowledgements"
        ]
      }
    ],
    "category": "CERT/CC Vulnerability Note",
    "csaf_version": "2.0",
    "notes": [
      {
        "category": "summary",
        "text": "### Overview\r\n\r\nVersions 1.1.5 and earlier of the mu HTTP daemon .....",
        "title": "Summary"
      }
    ],
    "publisher": {
      "category": "coordinator",
      "contact_details": "Email: cert@cert.org, Phone: +1412 268 5800",
      "issuing_authority": "CERT/CC under DHS/CISA https://www.cisa.gov/cybersecurity also see https://kb.
cert.org/ ",
      "name": "CERT/CC",
      "namespace": "https://kb.cert.org/"
    },
    "references": [
      {
        "url": "https://vuls.cert.org/confluence/display/Wiki/Vulnerability+Disclosure+Policy",
        "summary": "CERT/CC vulnerability disclosure policy"
      },
      {
        "summary": "CERT/CC document released",
        "category": "self",
        "url": "https://kb.cert.org/vuls/id/495801"
      },
      {
        "url": "https://derekabdine.com/blog/2022-arris-advisory",
        "summary": "https://derekabdine.com/blog/2022-arris-advisory"
      },
      {
        "url": "https://www.cisa.gov/uscert/ncas/tips/ST15-002",
        "summary": "https://www.cisa.gov/uscert/ncas/tips/ST15-002"
      }
    ],
    "title": "muhttpd versions 1.1.5 and earlier are vulnerable to path traversal",
    "tracking": {
      "current_release_date": "2022-08-05 20:02:52.605648+00:00",
      "generator": {
        "engine": {
          "name": "VINCE",
          "version": "1.50.3"
        }
      },
      "id": "VU#495801",
      "initial_release_date": "2022-08-04 18:22:24.069865+00:00",
      "revision_history": [
        {
          "date": "2022-08-05 20:02:52.605648+00:00",
          "number": "1.20220805200252.2",
          "summary": "Released on 2022-08-05 20:02:52.605648+00:00"
        }
      ]
    }
  }
}
```

```

    ],
    "status": "final",
    "version": "1.20220805200252.2"
  }
},
"vulnerabilities": [
  {
    "title": "The base firmware for this modem contains an MIT-licensed web server from an individual developer called \"muhttpd.\",",
    "notes": [
      {
        "category": "summary",
        "text": "The base firmware for this modem contains an MIT-licensed web server from an individual developer called \"muhttpd.\" This server has been unmaintained since 2010. The server has a path traversal vulnerability that allows any file on the modem to be read as root"
      }
    ],
    "cve": "CVE-2022-31793",
    "ids": [
      {
        "system_name": "CERT/CC V Identifier ",
        "text": "VU#495801"
      }
    ],
    "product_status": {
      "known_not_affected": [
        "CSAFPID-eb07f774-32d4-11ed-aeca-0aa659cdc35f"
      ]
    }
  }
],
"product_tree": {
  "branches": [
    {
      "category": "vendor",
      "name": "AT&T",
      "product": {
        "name": "AT&T Products",
        "product_id": "CSAFPID-eb07f774-32d4-11ed-aeca-0aa659cdc35f"
      }
    },
    {
      "category": "vendor",
      "name": "SaskTel",
      "product": {
        "name": "SaskTel Products",
        "product_id": "CSAFPID-eb082dc0-32d4-11ed-aeca-0aa659cdc35f"
      }
    }
  ]
}
}

```

Update vendor status

```
#update vendor status
api = f'https://kb.cert.org/vince/comm/api/case/{case}/vendor/statement/'
data = [{'vendor': 3548,
        'status': 'Not Affected',
        'references': ["http://www.test.gov", "https://www.google.com"],
        'share': True,
        'vulnerability': 'CVE-2020-19293',
        'statement': 'This is my statement'},
        {'vendor': 3548,
        'status': 'Affected',
        'statement': "Test",
        'references': ["http://www.test.gov", "https://www.google.com"],
        'share': True,
        'vulnerability': 'VU#785701.2'}]
r = requests.post(api, headers=headers, data=json.dumps(data))
print(r.text)
```

```
#update vendor status
api = f'https://kb.cert.org/vince/comm/api/case/{case}/vendor/statement/'
data = [{'vendor': 3548, # vendor ID only required if user belongs to multiple vendors in a case
        'status': 'Not Affected', # required: ['Affected', 'Not Affected', 'Unknown']
        'references': ["http://www.test.gov", "https://www.google.com"], # not required, must be a list
        'share': True, # not required, default = False
        'vulnerability': 'CVE-2020-19293', # required - must be in the form 'CVE-xxxx-xxxxx' or 'VU#xxxxxx.n'
        'statement': 'This is my statement'}] # not required
```

Look up CVE IDs (must have access to case)

```
# lookup CVE-2021-55555 - must have access to case otherwise 404
api = f'https://kb.cert.org/vince/comm/api/vuls/cve/2020-19293/'
headers={'content-type': 'application/json', 'Authorization': "Token {}".format(token) }
r = requests.get(api, headers=headers, stream=True)
print(r.text)
```

```
API: CVE Lookup: https://kb.cert.org/vince/comm/api/cve/2021-55555/

{  'case': {  'created': '2020-03-11T18:56:14.975973Z',
             'due_date': '2020-03-25T0000Z',
             'status': 'Active',
             'summary': 'This is a summary',
             'title': 'This is a title',
             'vuid': '123456'},
    'note': 'NOT Public',
    'vendors': [ {  'references': '',
                   'statement': '',
                   'statement_date': '2020-11-20T11:05:07.603524Z',
                   'status': 'Unknown',
                   'vendor': 'Test Vendor',
                   'vulnerability': 'CVE-2021-55555'}],
    'vulnerability': {  'cve': '2021-55555',
                       'date_added': '2020-03-11T20:37:51.629151Z',
                       'description': 'This is a description of the vulnerability',
                       'name': 'CVE-2021-55555'}}
```