
VU CAS NAUDOJIMO INSTRUKCIJA

TURINYS

I.	Apie VU CAS.....	2
II.	VU CAS veikimas. Administratoriaus atmintinė.....	3
III.	Tinklalo paruošimo darbui su VU CAS pavyzdys.....	4
1.	VU CAS bibliotekos diegimas.....	4
2.	Privačių duomenų apsaugojimas nuo tiesioginės prieigos.....	4
3.	Tinklalo paruošimas lankytojo atpažinimui.....	5
4.	Parengiamieji veiksmai, atliekami po sėkmingo lankytojo autorizavimo VU CAS.....	6
5.	Darbo pabaiga. Sesijų nutraukimas.....	8

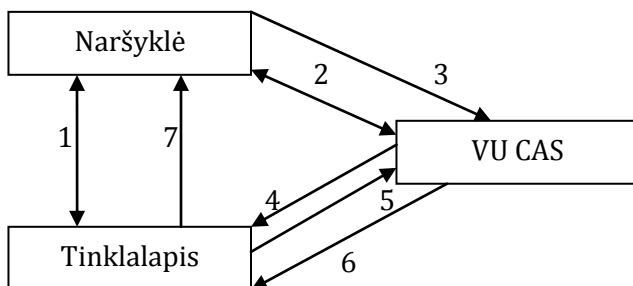
I. APIE VU CAS

CAS – Centrinė atpažinimo Sistema. CAS technologija plačiai naudojama visame pasaulyje. VU CAS – tai priemonė, leidžianti tinklalapio lankytojams pasiekti jiems skirtą, ribotos prieigos informaciją. Taip pat ši sistema suteikia galimybę tinklalapio lankytojo atpažinimą vykdyti ne lokaliai, o VU CAS serveryje (patikimumas, saugumas). Administratoriui nereikia atnaujinti naudotojų duomenų bazės, sinchronizuoti (paprastumas). VU CAS naudoja *https* – saugų duomenų perdavimo protokolą.

Tinklalapio administratoriaus prašymu, el. paštu perduodami darbai su VU CAS skirti duomenys – CAS biblioteka (*PHP* failai).

Žemiau pateikti žingsniai ir schema atvaizduojanti kaip vyksta atpažinimas.

1. Tinklalapio lankytojas naršyklėje atveria tinklalapį, kuriame nori pasiekti ribotos prieigos informaciją.
2. Tinklalapyje svečias randa nuorodą į VU CAS'ą, kuri atveria VU CAS tinklalapį.
3. VU CAS'e lankytojas suveda savo VU el. pašto prisijungimo duomenis: naudotojo vardą ir slaptažodį (toliau – ERID¹).
4. Lankytojui suvedus ERID, VU CAS'as vykdo atpažinimą ir tinklalapiui grąžina rezultatus.
5. Tinklalapis kreipiasi į VU CAS'ą su gautais rezultatais, prašydamas patvirtinti atpažinimą.
6. Sėkmingo atpažinimo atveju, VU CAS'as perduoda tinklalapiui informaciją apie naudotoją (VU el. pašto adresą, VU el. pašto prisijungimo vardą), nesėkmės atveju, grąžinamas klaidos pranešimas.
7. Priklausomai nuo gautų rezultatų, tinklalapis lankytojui naršyklėje atvaizduoja tam tikrą informaciją.



VU CAS panaudojimą galite išbandyti demonstraciniame tinklalapyje <http://cas-demo.vu.lt>. Kitame puslapyje veikimas aprašytas plačiau (skirta tinklalapio administratoriui).

¹ Elektroninės registracijos identifikatorius

II. VU CAS VEIKIMAS. ADMINISTRATORIAUS ATMINTINĖ

VU CAS naudotojų atpažinimui naudoja *LDAP* duomenų bazę. Kaip minėta ankstesniame paragrafe, VU CAS'as duomenų siuntimui/gavimui naudoja *https* protokolą. Tai reikalinga, nes atpažinimo metu siunčiami jautrūs duomenys: prisijungimo vardo ir slaptažodžio pora. Tinklaluiui tampa nereikalinga naudoti *https*, nes visa autorizacija atliekama nelokaliam. Žemiau pateikiami žingsniai, kaip vyksta informacijos siuntimas tarp tinklalapio ir VU CAS'o serverio.

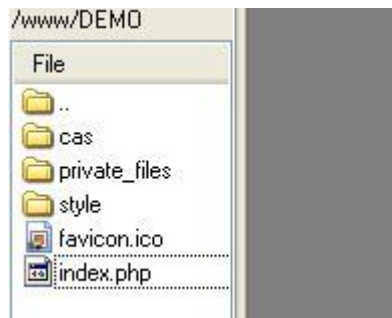
1. Lankytojas savo naršyklėje atveria tinklalapį, kuriame nori pamatyti ribotos prieigos duomenis. Kol lankytojas neprisijungęs, jam siūloma tai padaryti paspaudžiant nuorodą „Prisijungti per VU CAS“. Nuorodos tikslas (*target*) sugeneruojamas naudojant *CAS* tipo objekto metodą *generate_link()*; - *\$cas->generate_link()*;
2. Suvedęs savo ERID VU CAS'e po sėkmingo atpažinimo, lankytojas grąžinamas į tą pačią tinklalapio vietą, iš kurios vyko nukreipimas.
3. Tinklapis atskiria, ar įvyko autentifikacija pagal *\$_GET['lt']* reikšmę. Tai – prisijungimo bilietas (ang.: „*login ticket*“). Jei *\$_GET['lt']* neegzistuoja arba yra tuščias, tokiu atveju lankytojas liko neatpažintas.
4. Jei *\$_GET['lt']* egzistuoja, tuomet galima kreiptis į VU CAS'ą dar kartą, prašant pateikti informaciją apie lankytoją. Tai padaroma naudojant naudojant *CAS* tipo objekto metodą *validate()*; - *\$cas->validate()*; Tokios išraiškos rezultatas – masyvas su lankytojo VU el. pašto adresu (be galūnės “.vu.lt”) ir prisijungimo vardu. Jei *validate()* grąžina “false”, tuomet atpažinimas neįvyko. Prisijungimo vardas bus reikalingas atsijungiant nuo VU CAS sistemos, tad jį reikia saugoti.
5. Tikrinamos lankytojo teisės (galima tiek pagal prisijungimo vardą, tiek pagal dalį el. pašto adreso) ir atitinkamai rodoma informacija.
6. Lankytojui baigus darbą vykdomas atsijungimas VU CAS sistemoje: *\$cas->logout(\$login)*, čia *\$login* – VU el. pašto naudotojo vardas.

III. TINKLALAPIO PARUOŠIMO DARBUI SU VU CAS PAVYZDYS

1. VU CAS BIBLIOTEKOS DIEGIMAS

Tinklalapio administratoriaus prašymu, el. paštu perduodami darbui su VU CAS skirti duomenys – VU CAS biblioteka (*PHP* failai). Ji turi būti įkelta į Jums skirtus resursus. Rekomenduojama laikyti juos atskirame kataloge.

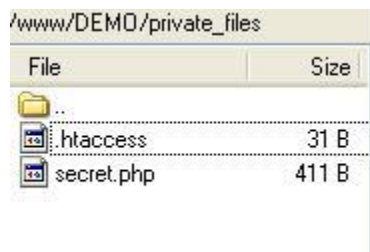
Žemiau pateiktame pavyzdyje VU CAS biblioteka laikoma /cas kataloge.



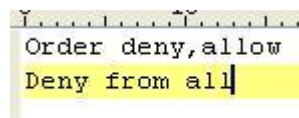
1pav. Šakninis katalogas

2. PRIVAČIŲ DUOMENŲ APSAUGOJIMAS NUO TIESIOGINĖS PRIEIGOS

1-ame pav. matomas katalogas /private_files. Jame slypi ribotos prieigos informacija, kuri bus prieinama tik tam tikrai grupei žmonių. Žemiau pateikiamas pavyzdys, kaip galima uždrausti pasiekti failus tiesiogiai, t.y. vedant kelią iki failo tiesiai naršyklės adresu juostoje – tam skirtas *.htaccess* failas.



2pav. “/private_files” katalogas



3pav. “.htaccess” failo turinys

This error (HTTP 403 Forbidden) means that Internet Explorer was able to connect to the website, but it does not have permission to view the webpage.

For more information about HTTP errors, see Help.

4pav. IE8 rodomas pranešimas, adresų juostoje suvedus tiesų kelią iki “/private_files” katalogo

3. TINKLALAPIO PARUOŠIMAS LANKYTOJO ATPAŽINIMUI

Norint nukreipti lankytoją į VU CAS, reikalinga sugeneruoti nuorodą į šią sistemą. Žemiau pateikiama kodo iškarpa kaip tai padaroma demonstraciniame VU CAS panaudojamumo tinklalapyje. Šiame pavyzdyje informacijos apie CAS prisijungimą saugojimui naudojami PHP sesijos kintamieji ($$_SESSION$).

```
<!-- Startuojam sesiją, įtraukiam cas'o biblioteką, bandom autorizuoti naudotoją -->
<?php
    session_start();

    include_once("cas/cas.php");
    $cas = new CAS();

if (isset($_GET['lt']) && empty($_SESSION)) {
    if ($info = $cas->validate()) {
        $_SESSION["email"] = $info["email"];
        $_SESSION["login"] = $info["user"];
    }
    else echo $cas->error;
}
?>
<!-- ////////// -->

<div class="title">
VU CAS demonstracinė versija
</div>
<br><br>

<!-- Vieša informacija -->
<div class="information">
<?php if (!isset($_SESSION["email"])) { ?>

Norėdami peržiūrėti savo duomenis - prisijunkite.

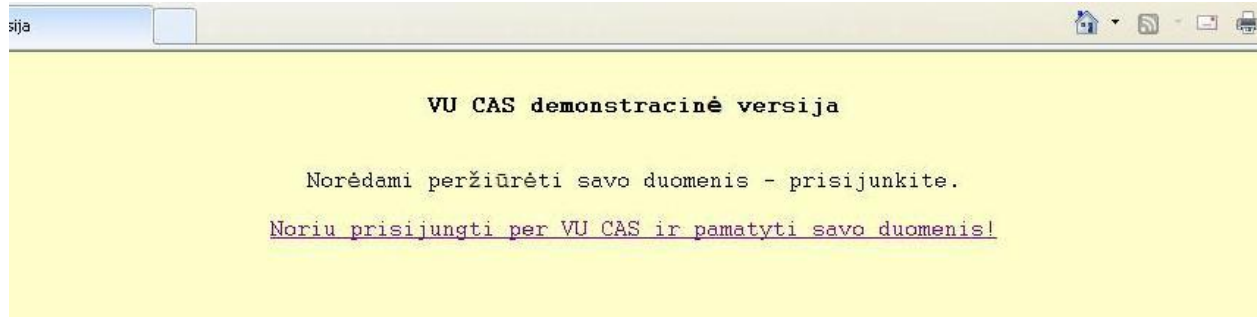
<br><br>
<?php } ?>
<!-- ////////// -->

<!-- Jei naudotojas neprisijungęs, rodomė prisijungimo nuorodą. -->
<?php if (!isset($_SESSION["email"]) || isset($_GET["logout"])) { ?>

<a href="<?php echo $cas->generate_link(); ?>">Noriu prisijungti per VU CAS ir pamatyti savo duomenis!</a><br>
<?php } ?>
<!-- ////////// -->

</div>
```

5pav. *index.php* failo kodo iškarpos. Pažymėtuose blokuose atliekami veiksmai, reikalingi sugeneruoti teisingą nuorodą į VU CAS



6pav. *index.php* matomas naršyklėje prieš lankytojo prisijungimą

4. PARENGIAMIEJI VEIKSMAI, ATLIEKAMI PO SĖKMINGO LANKYTOJO AUTORIZAVIMO VU CAS

Parengiame privačius, tam tikrai naudotojų grupei skirtus duomenis. Tarkime, norime parodyti informaciją, kurią gauname iš VU CAS po sėkmingos autorizacijos (grįžimas iš VU CAS vyksta į tą vietą, iš kurios buvo kreiptasi. Šiuo atveju tai bus *index.php* failas) (8pav.). Atlikus atpažinimą, objekte *\$info* (šiuo atveju) saugomi šie naudotojo duomenys: prisijungimo prie VU el. pašto vardas, el. pašto adresas be galūnės „.vu.lt“ (7pav.).

```
<!-- Startuojam sesiją, įtraukiam cas'o biblioteką, bandom autorizuoti naudotoją -->
<?php
    session_start();

    include_once("cas/cas.php");
    $cas = new CAS();

    if (isset($_GET['lt']) && empty($_SESSION)) {
        if ($info = $cas->validate()) {
            $_SESSION["email"] = $info["email"];
            $_SESSION["login"] = $info["user"];
        }
        else echo $cas->error;
    }
?>
<!-- ////////// -->
```

7pav. iškarpa iš *index.php* failo. Masyve *\$info* saugomi duomenys išsaugojami *\$_SESSION* masyve, kad išliktų iki pat sesijos pabaigos

```

<!-- Jei naudotojas prisijungęs, rodome jam skirtą informaciją -->
<?php
    if (!empty($_SESSION["email"])) {
?>
<div class="private_info">
    <?php include("private_files/secret.php"); ?>
</div>
<?php } ?>
<!-- ////////// -->

```

8pav. iškarpa iš *index.php* failo. Jei autorizavimas sėkmingas, t.y. jei sesijos masyve egzistuoja el. pašto adresas, tuomet rodomi "*private_files/secret.php*" faile esantys duomenys

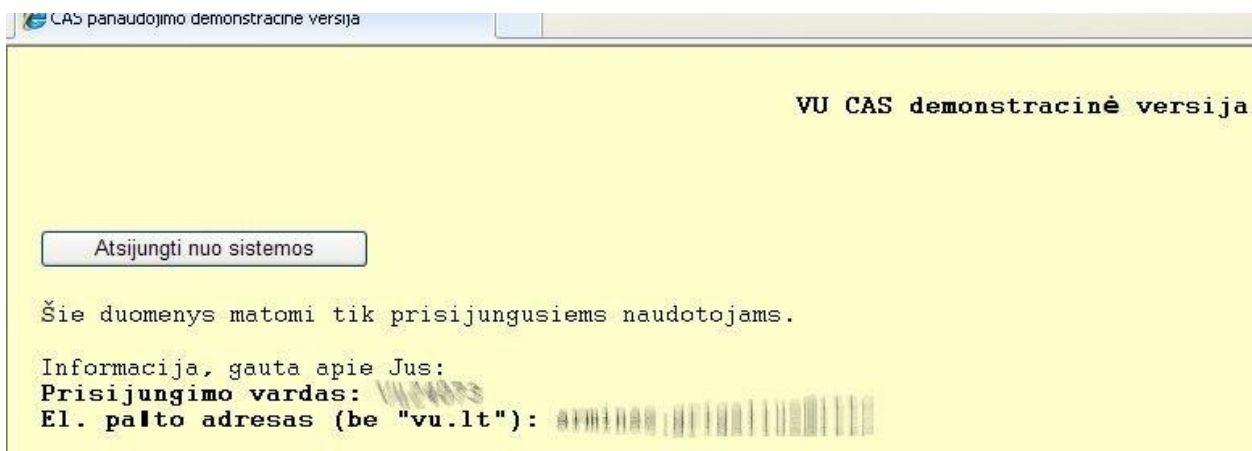
Žemiau pateikiamas */private_files/secret.php* failo kodas ir kaip jis atvaizduojamas po sėkmingo atpažinimo (jei sėkmingai pavaizduotas 8-ame pav.)

```

Šie duomenys matomi tik prisijungusiems naudotojams.<br /><br />
Informacija, gauta apie Jus:<br />
<?php
    echo "<b>Prisijungimo vardas: </b>". $_SESSION["login"] . "<br />";
    echo "<b>El. pašto adresas (be \"vu.lt\") : </b>". $_SESSION["email"] . "<br />";
?>
<br /><br />

```

9pav. "*private_files/secret.php*" failo kodas



10pav. "*/private_files/secret.php*" failo atvaizdavimas po sėkmingo atpažinimo

5. DARBO PABAIGA. SESIJŲ NUTRAUKIMAS

Svarbu žinoti, jog šiame pavyzdyje minimos dvi skirtingos sesijos – tinklalapio ir VU CAS' o. VU CAS'e esanti sesija nutraukiama naudojant CAS tipo objekto metodą *logout(login)*; , kurio parametras yra lankytojo prisijungimo vardas.

\$_SESSION masyve esančios reikšmės saugomos lokaliai. Jomis galima manipuluoti pagal poreikį. Rekomenduojama abi minėtas sesijas nutraukti kartu.

Žemiau pateikiamas pavyzdys, kaip ir kada sugeneruojamas atsijungimo mygtukas. „*echo \$cas->error;*“ nesėkmingo atsijungimo atveju ekrane atspausdina klaidos pranešimą – tai padeda lengviau susekti programavimo spragas.

```
<!-- Baigus darbą, reikia išvalyti sesijos kintamuosius, bei pranešti CAS'ui, kad naudotojas atsijungė -->
<?php
if (isset($_GET["logout"]) && isset($_SESSION["email"])) {
    if ($cas->logout($_SESSION["login"])) {
        echo "Sėkmingai atsijungėte nuo sistemos<br />";
        unset($_SESSION);
        session_destroy();
    }
    else echo $cas->error;
}
?>
<!-- ////////// -->

<!-- Jei naudotojas prisijungęs, rodome atsijungimo mygtuką -->
<?php
if (!empty($_SESSION["email"])) {
?>

<div class="user_info">
    <form name="logout" action="index.php" method="GET">
    <input type="submit" name="logout" value="Atsijungti nuo sistemos" />
    </form>
</div>

<?php
}
?>
<!-- ////////// -->
```

11pav. *index.php* failo kodo iškarpa. Atsijungimo mygtukas generuojamas tik tuomet, kai lankytojas prisijungęs. *\$cas->logout(\$_SESSION["login"])* sėkmingo atsijungimo metu grąžina *true*, kitu atveju atspausdinamas klaidos pranešimas